DTU Library

A detailed emission inventory of nitrogen oxides for Denmark

Runge, E.H.; Asman, W.A.H.; Kilde, N.A.

Publication date:
1991

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Runge, E. H., Asman, W. A. H., & Kilde, N. A. (1991). *A detailed emission inventory of nitrogen oxides for Denmark*. Risø-M No. 2929

A detailed Emission Inventory of Nitrogen Oxides for Denmark

Risø-M-2929

*Erik H. Runge, *Willem A.H. Asman and
**Niels A. Kilde

*National Environmental Research Institute (NERI)
Division of Emissions and Air Pollution
Frederiksborgvej 399
DK-4000 Roskilde, Denmark

**Risø National Laboratory
Systems Analysis Department
DK-4000 Roskilde, Denmark

Risø National Laboratory, Roskilde, Denmark
August 1991

Abstract The Danish NO_x emissions from domestic heating, energy generation, industry, road traffic, point sources and maritime vessels have been distributed on a 1x1 km^2 grid and on municipalities.

The total NO_x emission is calculated to 294 ktonne NO_2 a^{-1}.

This report describes the distribution, lists the computer programs used and gives the format of the final storage file.

ISBN 87-550-1730-4
ISSN 0418-6435

Grafisk Service, Risø, 1991

Contents

Summary

The emissions of NO_x in Denmark from domestic heating, energy generation, industrial combustion, industrial production, road traffic, large point sources and some maritime vessels have been computed and distributed on a grid with 1x1 km^2 elements. The land based area sources have also been distributed among municipalities.

Emission data from the EEC database Corinair for Denmark have been used for domestic heating, energy generation, industrial combustion and industrial production. For road traffic the emissions have been calculated on the basis of distribution of the traffic and emission factors as used in Corinair. The distribution of the emissions, from road traffic, over the country have been done using data from the Road Data Laboratory (Vejdatalaboratoriet). Finally, for freighters and ferries information from the Danish State Railways (DSB) and emissions proposed by the European Monitoring and Evaluation Progam (EMEP) have been used.

The distribution of the emissions from area sources on land over the 1x1 km^2 grid has been performed using statistics on the total number of inhabitants in each municipality, inhabitants in urban areas and the registration of the land use by Runge and Asman (1989). Further, the distribution of emissions among municipalities has been done using both the grid and the land use registrations.

The total Danish NO_xemission, expressed as NO_2, is estimated to be 294 ktonne NO_2 a^{-1}.

Finally, in Appendix A, the source code is listed of the main computer programs used in the calculations.

Sammenfatning

Emissionen af NO_x i Danmark fra boligopvarmning, energi produktion, industriel forbrænding, industriel produktion, vejtrafik, punktkilder samt nogle skibs- og færgeruter er opgjort og fordelt på et 1x1 km^2 net. Arealkilderne på land, er desuden fordelt på kommuner.

For boligopvarmning, energi produktion, industriel forbrænding og industriel produktion er brugt data fra EF databasen Corinair, for Danmark. For vejtrafik er emissionen beregnet på baggrund af fordeling af trafikken og emissionsfaktorer som i Corinair. Fordelingen af emissionen fra vejtrafik over landet er sket efter data fra Vejdatalaboratoriet. Endelig for skibs- og færgeruter er brugt oplysninger fra DSB og emissions størrelser som er foreslået af EMEP (European Monitoring and Evaluation Progam).

Fordelingen af emissionen fra arealkilder på land, over 1x1 km^2 nettet er foretaget udfra befolkningstallene i de enkelte kommuner totalt og i byområder samt udfra registreringen af arealets anvendelse som opgjort i »Land-use« registreringen (Runge og Asman, 1989).

Udfra opgørelsen af emissionerne på 1x1 km^2 er, ved hjælp af registreringen i »Land-use« af hvilke kommuner hver enkelt km^2 ligger i, beregnet emissionens fordeling på kommuner.

Den samlede danske NO_x emission, udtrykt som NO_2 er beregnet til 294 ktonne NO_2 $år^{-1}$.

Endelig er i Appendix A trykt kildeteksten til de væsentligste edb-programmer, der er blevet brugt i beregningerne.

1 Introduction

The deposition of some nitrogen components has unfortunate effects on land ecosystems (Nilsson and Grennfelt, 1988) and on marine ecosystems. Nitrogen oxides (NO_x), the sum of nitric oxide (NO) and nitrogen dioxide (NO_2) comprise one of the most influential categories of these nitrogen components. Therefore upon request of the Danish Environmental Protection Agency, the National Environmental Rersearch Institute (NERI) undertook to compute concentration patterns and import/export balances of NO_x for Denmark and adjacent waters. For this purpose, a more detailed emission inventory was required than currently available. The present report describes the development and characteristics of such a detailed emission inventory for NO_x.

A NO_x emission inventory for Denmark anno 1985 had already been made by the System Analysis Department of Risø National Laboratory, Roskilde, Denmark, within the framework of the EEC Corinair project. This emission inventory consists of two parts: detailed information on the emissions from large point sources and information on emissions from area sources for each county (»amt« in Danish). The latter information, however, is not detailed enough to be used for modelling gradients over Denmark. It was therefore decided to make a more detailed emission inventory for area sources, partly by using information from the existing inventory on domestic heating, energy generation and industrial sources. These emissions were then distributed among municipalities (»kommuner« in Danish) and on a grid with 1x1 km^2 elements applying the land-use database for Denmark, developed at NERI (Runge and Asman, 1989). Emissions from road traffic and maritime vessels were computed separately using detailed data from the Danish Road Data Laboratory (RDL) (»Vejdatalaboratoriet« in Danish) under the Road Directorate and from the Danish State Railways (DSB).

The grid used in this inventory is the Universal Transverse Mercator (UTM) grid (Geodætisk Institut, 1981). The UTM grid is divided into zones. Most of Denmark is situated in Zone 32, but the eastern part of Zeeland and Bornholm are situated in Zone 33. To make the inventory more uniform, the Zone 33 coordinates of Zeeland and Bornholm, have been transformed to Zone 32 coordinates. That is, all coordinates in the inventory are given in UTM Zone 32.

Risø National Laboratory computed the emission from domestic heating, energy generation and industrial production and combustion.

Erik Runge at NERI performed the computations of emissions from road traffic and maritime vessels, the distribution of NO_x emissions among municipalities and grid elements and the construction of the detailed inventory.

The report is divided into four main chapters: (1) Area sources on land, (2) large point sources on land, (3) emissions from maritime vessels and (4) total emissions. This division was chosen to distinguish between area and point sources and further to relate the land based emissions to municipalities.

2 Emissions from area sources on land

2.1 Introduction

The following categories of NO_x emissions from area sources have been taken from the Corinair database:

- domestic heating
- energy generation
- industrial combustion
- industrial production.

The difference between emissions from industrial combustion and those from industrial production is that in combustion, the flue gases are not in direct contact with the produced material, whereas in production they are.

The emissions from road traffic have not been taken from the Corinair database, but have been calculated on the basis of information from the Road Data Laboratory (RDL) (S.Schrøder, RDL, pers. comm., 1990). From the RDL we have received information on the total traffic in 1988 and the location of, and traffic on major roads in Denmark.

2.2 Computation of the gridded emissions

In the Corinair database the total emission from domestic heating, energy generation, industrial combustion and industrial production for each county is given. For use in dispersion models, we wanted to have the emissions on a regular grid. Therefore the Corinair data were distributed on a 1x1 km^2 grid.

But as dispersion models using a 1x1 km^2 grid may be considered to be unnecessarily time consuming, the emissions were additionally distributed on a 5x5 km^2 grid.

Within each county and municipality the emissions were distributed according to the land-use registration of Runge and Asman (1989), where for each km^2 it has been registered i.a. how much of the area is urban and how much is farmland.

It was assumed that all industry and energy generation is placed in urban areas. The total emission from industry and energy generation within a county was distributed evenly over all areas which have been registered as urban.

The emissions from domestic heating were within the urban areas distributed according to the number of inhabitants in urban areas, while in other areas they were distributed evenly according to the total number of inhabitants not living in urban areas. The distribution on the 1x1 km^2 grid was conducted using again the land-use registration.

The program, NOX-CORINE was used for this distribution of emissions from domestic heating, energy generation, industrial combustion and industrial production. The program is listed in Appendix A.

The emissions from the road traffic have also been distributed on the 1x1 km^2 grid. From the RDL we have received information on the positions of all Danish state and county roads, i.e. all major roads in Denmark. Each road was divided into smaller pieces. To each piece were attached i.a. the road number, the coordinates of the beginning and ending points and the yearly traffic.

Using questionnaires the RDL has estimated the total traffic per year on all Danish roads. By subtracting the sum of the traffic on main roads from the total traffic, we obtained an estimate of the traffic on minor (mainly municipal) roads in Denmark.

The emissions from the road traffic are calculated on the basis of a distribution of the traffic and average emission factors computed with the Copert model of Corinair (Eggleston et al, 1989). Traffic data and emission factors are listed in Table 1. It follows i.a. from this table that the total emission from passenger cars and light vans is of the same magnitude as that from diesel vehicles over 3500 kg.

As we for most of the major roads have their placement in the country by their beginning and ending coordinates for each road piece, they have in this way been distributed on the 1x1 km^2 grid net. For road pieces where coordinates of the beginning and ending points were missing, the emissions were distributed evenly over the county in which the road is situated.

The emissions on minor roads have been distributed over the municipalities according to the number of inhabitants in each municipality (1988 statistics given by the Danish Statistical Office). Within municipalities the emissions have been distributed on the 1x1 km^2 grid, using information in the land-use registration and population data from the Danish Statistical Office. We have, using this distribution key, assumed that all inhabitants in Denmark drive equally much on municipal roads.

The program ROAD-TRAFFIC was used for this distribution. It is listed in Appendix A.

Thus all land based NO_x area emissions were computed and distributed on a 1x1 km^2 grid. The data were stored in a data file named NOX-AREA. The distribution of emissions over municipalities was done using the file NOX-AREA and the land- use registration, the latter in which each km^2 is subordinated to municipalities. The results of this distribution are shown in Table 2.

Figures 1 to 5 show the geographical distribution of the NO_x emissions for the five different categories; domestic heating, energy generation, industrial combustion, industrial production and traffic. Figure 6 shows the geographical distribution of the total NO_x emission from area sources in Denmark.

3 Emissions from large point sources on land

3.1 Introduction

The following categorizing of NO_x emission point sources have been taken from the Corinair database:

- power stations over 300 MW
- oil refineries
- manufacturing of sulphuric and nitric acid
- iron- and steelworks
- paper mills
- car lacquering works with over 100000 cars a^{-1}.

Of these categories it is mainly major power stations which in Denmark are categorized as point sources. Sources, in Denmark, which are too small to be categorized as a point source are registred in the Corinair database for area sources.

3.2 Location of the emissions

In the Corinair database the location of a point source is given by its global longitude and latitude. These have been transformed to UTM Zone 32 coordinates.

In Table 3, are listed the total NO_x emissions from large point sources for the counties in Denmark. In Figure 7 are plotted the major point sources. Note that the diameter of a circle in the figure shows the relative magnitude of the corresponding source or cluster of sources.

4 Emissions from maritime vessels

4.1 Introduction

The emissions from maritime vessels can be divided into different categories, such as:

- international freighter traffic
- national freighter traffic
- international ferry traffic
- national ferry traffic
- fishing boats
- leisure vessels
- military ships
- other maritime vessels.

In this study only emissions from international freighter traffic and some of the main national ferries were included.

Of importance for the NO_x emissions are the engine type of the ship (i.a. whether two- or four-stroke), the average and maximum speed, the engine maximum speed, the engine power used, the size of the ship, and especially for ferries the time used in port and the time used for arrival at and departure from ports. However, these factors have not been taken into account in this investigation.

4.2 NO_x emissions from international freighter traffic

There are two main routes for international freighter traffic through Danish waters. These are (see Figure 8):

Route T from Skagen, east of the island of Anholt, through the Great Belt (Storebælt), south of Lolland and Falster and into the Baltic Sea (Østersøen).

Route D/B which initially is as route T, but from Anholt it goes southeast through the Sound into the Baltic Sea.

North of Skagen and within the Baltic Sea the ships sail in different directions depending on their destinations. For this reason, the NO_x emissions have been estimated and gridded only for those route sections as shown in Figure 8.

There are other freighter routes in Danish national waters, but these two are the most important.

The total emission was calculated using the emissions given by Bremnes (1990). For route T from Skagen to southeast of Lolland, Bremnes gives an emission of 34 tonne NO_x per nautical mile per year (tonne nmi^{-1} a^{-1}). South east of Lolland a different route passing through the Kiel Canal joins route T, and the emission from the joining point further eastwards is 86 tonne nmi^{-1} a^{-1}. Along route D/B the emission is 21 tonne nmi^{-1} a^{-1}. These emissions must be taken as average emissions for all types of freighters.

4.3 NO_x emissions from national ferry services

The ferry services in Denmark are run by different companies, of which the largest are DSB, DFDS and Bornholmstrafikken. Of these, DSB is by far the largest, having 29 ferry routes. We were given information on the 9 most important DSB routes, including their yearly oil consumption. The 9 routes are shown in Figure 9.

The NO_x emissions were calculated on the basis of the oil consumption, using an emission factor of 70 g NO_x per kg fuel, as given by Bremnes (1990). The yearly oil consumption was given in m^3. As the density of the fuel used varied between 0.80 and 0.95 tonne m^{-3}, a mean density of 0.85 tonne m^{-3} was used.

For each of the 9 routes the emission was calculated and distributed evenly along the route (thus not taking into account the time the ferries spent in port or navigating in and out of ports).

4.4 Computation of the gridded emissions

The estimated NO_x emissions from international freighter traffic and main ferry services are shown in Table 4.

The emissions from international freighter traffic as distributed on the UTM grid are shown in Figure 10. The gridded emissions from main ferry services are shown in Figure 11. Within the Sound, the gridded NO_x emissions from main ferry services are shown in more detail in Figure 12.

The two main programs used in the calculations of NO_x emissions from maritime vessels are SHIPS for freighters, and FERRY for main ferry services. They are listed in appendix A.

5 Total emissions

The total Danish NO_x emission (expressed as NO_2), comprising all major point and area sources over land and sea, as estimated in this survey, adds up to approximately

294 ktonne $NO_2\ a^{-1}$.

In table 5 are listed the total NO_x emissions from the different source categories in this survey, and in Figure 13 these are visualised in a pie chart.

6 Comparison with other emission estimates and uncertainties in the inventory

6.1 Comparison with other emission estimates

Several NO_x emission inventories have been made by various researchers for various source categories, prior to the present inventroy, but as there have been no standards within this field of work, comparison between inventories is difficult.

For the total NO_x emission from road traffic, called the road traffic emission in the following, the comparison is most straightforward.

Torp (1983) has made an inventory for the year 1980 in which this emission is estimated to be

75000 tonne $NO_2\ a^{-1}$.

At the Technical University of Denmark, Nielsen et al. (1988) has found the road traffic emission in 1986 to be

83000 tonne $NO_2\ a^{-1}$.

In the Corinair database the total NO_x emission from road transport is

102831 tonne $NO_2\ a^{-1}$.

In the present survey, the road traffic emission is estimated to be

101775 tonne $NO_2\ a^{-1}$.

As we have used the same emission factors as in Corinair our figure is similar to that given by Corinair. No greater precision could be expected as the average emission factors used have only two digests.

Generally it can be stated that for the road traffic emission, our estimate is not very different from the others mentioned.

The total NO_x emission from point and area sources, but not including emissions from maritime vessels, has been calculated by Torp (1983) to be

253000 tonne $NO_2\ a^{-1}$.

If in the present survey the total emissions from area sources and point sources are added together, we get a figure of

247000 tonne $NO_2\ a^{-1}$

which is very similar to that found by Torp (1983).

Torp (1983) estimated the total NO_x emission from power stations to be

122000 tonne $NO_2\ a^{-1}$.

In the present study, the total emission from large point sources is found to be

131000 tonne $NO_2\ a^{-1}$.

Slightly greater, indicating that the power stations are responsible for most of the NO_x from the point source category.

No other published estimates of NO_x emissions from maritime vessels in Danish waters have been found. An unpublished estimate by Tom Wismann (dk Teknik, pers. comm., 1990) of the total NO_x emission from maritime vessels is

29000 tonne $NO_2\ a^{-1}$

which is very similar to the

24330 tonne $NO_2\ a^{-1}$

found in the present investigation.

The somewhat higher figure of Wismann (1990) can be explained by the fact that we have omitted a great number of maritime transports from our investigation.

6.2 Uncertainties in the inventory

It is estimated that the uncertainties on emission estimates in inventories using emission factors is > 20% standard deviation (Baars, 1990). What we estimate, is also the case in the present investigation.

The distribution of area sources over municipalities has in the present study been done according to the number of inhabitants in each municipality. Other distribution keys could have been used, e.g. for traffic or for domestic heating and energy generation. Which would lead to different results. Thus the resulting distribution, as listed in Table 2, should be treated with caution.

The uncertainty in the estimate of the total emission from maritime vessels is very large, mainly because a great number of lesser transport were omitted. The real total emission could well be twice as high.

Acknowledgements

This detailed emission inventory was made within the framework of the projects »NPo« and »Marine Research Programme '90«, which were both supported financially by the Danish Agency of Environmental Protection.

We wish to thank Svend Schrøder from the Road Data Laboratory for making road traffic data available, Bo V. Thomsen from DSB for giving information on the main ferry services and Nicky Brown, NERI for proofreading.

References

Baars, Hans-Peter (1990). Accucary of emission inventories. In: Racyna, J.M. (1990). Proc. of the workshop on international emission inventories. Regensburg, FRG. Norwegian Institute for Air Research, Lillestrøm, Norway.

Bremnes, P.K. (1990). Calculations of exhaust gas emissions from sea transport. Methodology and results. Proc. EMEP workshop on Emissions from Ships, 7-8 June 1990, Oslo, Norway, State Pollution Control Authority, Oslo, Norway.

Eggleston, H.S., N. Gorissen, R. Joumard, R.C. Rijkeboer, Z. Samaras, K.H. Zierock. (1989). Corinair working group on emission factors for calculating 1985 emissions from road traffic. Volume 1: Methodology and emission factors, Environment and quality of life. Final Report EUR 12260 EN. Commission of the European Communities, Brussels, Luxembourg.

Geodætisk Institut (1981). UTM nettet, opbygning og anvendelse. (The UTM Grid, construction and use; in Danish). 4th edition, Geodætisk Institut, Copenhagen, Denmark.

Nielsen, A.S., P.H. Pedersen, J.Schramm, S.C. Sorenson (1988). Totalemissionen af CO, HC, NO_x og partikler fra bilparken i Danmark. En prognose fra 1980 til 2010, (Total emissions of CO, HC, NO_x and particles from the Danish carpark. A forecast from 1980 to 2010; in Danish). Report RE 88-5, The Technical University of Denmark, Laboratory for Energetics, Lyngby, Denmark.

Nilsson, J. and P. Grennfelt (1988). Critical loads for sulphur and nitrogen. Report 1988:15, Nordic Council of Ministers, Copenhagen, Denmark.

Runge, E.H. and W.A.H. Asman (1989). Land-use in Denmark, DMU LUFT-B135, Departement of Emissions and Air Pollution. National Environmental Research Institute, Roskilde, Denmark.

Schramm, J. and S.C. Sorenson (1991). BILEMIS. En prognosemodel til beregning af køretøjsemissioner, (CAREMIS. A forcasting model for computation of emissions from vehicles; In Danish). Report RE 91-1, The Technical University of Denmark, Laboratory for Energetics, Lyngby, Denmark.

Torp, U. (1983). Emissionsoversigt for SO_2 og NO_x i Danmark, 1980. (Emission survey for SO_2 and NO_x in Denmark, 1980; in Danish). Report MST LUFT-A80, National Agency of Environmental Protection, Air Pollution Laboratory, Roskilde, Denmark.

Tables

Note that in several of the tables, the NO_x emissions are listed with greater accuracy than warranted by the uncertainties in the present investigation. Also note that the emissions are expressed as NO_2.

Table 1. Yearly road traffic in Denmark and NO_x emission factors for different categories of vehicles.

The data are based on the report of Eggleston et al. (1989) and computed with the Copert model of Corinair.

	Total traffic in 1988 (Bill. km)	Emission factors (g NO_2 km^{-1})	NO_x Emission (Ktonne a^{-1})
Passenger cars and light vans, petrol	22.14	2.1	46.49
Heavy vans, petrol	0.44	6.0	2.64
Diesel vehicles under 3500 kg	6.55	0.67	4.39
Diesel vehicles over 3500 kg	2.78	17.39	48.34
Total			101.86

Table 2. NO_x emissions for different source categories and emission densities for each municipality and county in Denmark. Only area sources are listed in this table. Point sources are for each county listed in Table 3.

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
101 København	764.4	313.7	1126.6	137.5	3990.6	6332.7	71.8
147 Frederiksberg	132.2	52.3	194.8	23.8	633.1	1036.1	118.1
013 København and Frederiksberg	896.6	366.0	1321.3	161.2	4623.7	7368.8	76.0

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
151 Ballerup	75.4	89.8	111.2	13.6	845.4	1135.5	33.3
153 Brøndby	37.0	68.0	84.1	10.2	1030.3	1249.7	60.3
155 Dragør	20.4	24.3	30.1	3.7	154.1	232.7	12.8
157 Gentofte	112.6	126.9	166.1	20.2	1100.8	1526.7	59.8
159 Gladsakse	102.4	118.3	151.0	18.4	1320.7	1710.8	68.4
161 Glostrup	34.1	40.6	50.3	6.1	573.5	704.6	52.9
163 Herlev	44.4	52.3	65.5	8.0	478.7	648.9	53.9
165 Albertslund	47.0	56.0	69.3	8.4	583.0	763.6	33.1
167 Hvidovre	82.1	94.8	121.1	14.7	867.5	1180.2	54.5
169 Høje Tåstrup	72.7	86.5	107.1	13.0	1197.7	1477.0	18.6
171 Ledøje-Smørum	14.5	17.3	21.4	2.6	107.3	163.1	5.2
173 Lyngby-Tårbæk	80.4	95.8	118.6	14.4	916.7	1225.9	31.5
175 Rødovre	60.2	67.7	88.7	10.8	744.2	971.6	80.2
181 Søllerød	51.8	61.8	76.6	10.0	682.6	882.8	22.2
183 Ishøj	33.3	39.9	49.6	6.0	565.4	694.3	28.1
185 Tårnby	69.5	76.3	102.5	12.5	521.4	782.1	12.5
187 Vallensbæk	19.5	23.3	28.8	3.5	376.8	451.8	49.2
189 Værløse	29.9	35.5	44.0	5.4	402.4	517.2	15.2
015 Københavns amtskommune	1007.0	1175.2	1485.9	181.7	12468.5	16318.4	31.0

Table 2 continued

No. Municipality	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	Emission density tonne NO_2 km^{-2} a^{-1}
	Emission (tonne NO_2 a^{-1})						
201 Allerød	31.0	40.0	51.7	21.2	483.9	627.9	9.3
205 Birkerød	31.1	40.1	51.8	20.6	534.0	677.5	20.2
207 Farum	24.1	31.1	40.3	16.5	347.5	459.5	20.3
208 Fr.borg-Humlebæk	27.0	34.8	45.1	18.4	407.0	532.4	7.4
209 Frederikssund	24.0	31.1	40.2	6.3	272.9	384.4	9.4
211 Frederiksværk	25.6	33.2	42.9	17.5	319.9	439.1	4.9
213 Græsted-Gilleje	25.3	32.8	42.4	17.3	327.7	445.5	3.3
215 Helsinge	25.1	32.3	41.8	17.1	340.6	457.0	3.1
217 Helsingør	81.3	105.1	136.0	55.6	868.7	1246.6	10.3
219 Hillerød	48.6	63.0	81.5	33.3	710.7	937.0	7.1
221 Hundested	12.6	16.3	21.1	8.6	119.1	177.7	5.6
223 Hørsholm	34.1	44.0	56.9	23.1	458.7	616.8	19.7
225 Jægerspris	11.1	14.4	18.6	7.6	192.5	244.2	2.6
227 Karlebo	27.2	35.2	45.5	18.6	366.7	493.2	12.3
229 Skibby	8.4	10.8	14.0	5.7	74.3	113.4	1.4
231 Skævinge	7.4	9.3	12.1	4.9	148.4	182.1	2.7
233 Slangerup	11.1	14.4	18.6	7.6	185.4	237.1	5.2
235 Stenløse	18.4	23.9	30.9	12.5	237.4	323.1	4.9
237 Ølstykke	18.6	24.2	31.4	12.6	226.0	312.8	10.7
020 Frederiksborg amtskommune	492.2	635.9	822.6	335.1	6621.6	8907.3	6.6

No. Municipality	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	Emission density tonne NO_2 km^{-2} a^{-1}
	Emission (tonne NO_2 a^{-1})						
251 Bramsnæs	11.0	14.9	20.5	2.2	326.1	374.7	4.7
253 Greve	64.5	86.9	119.5	12.8	1054.4	1338.2	22.2
255 Gundsø	17.6	23.7	32.6	4.0	174.4	252.4	4.0
257 Hvalsø	10.1	13.6	18.8	2.0	194.1	238.6	3.3
259 Køge	51.0	69.2	95.4	10.2	1284.9	1510.8	12.2
261 Lejre	11.9	16.0	22.1	2.4	402.8	455.2	5.2
263 Ramsø	12.2	16.7	23.0	2.4	110.5	164.9	2.4
265 Roskilde	69.4	94.0	129.5	13.8	1013.0	1319.6	16.4
267 Skovbo	18.6	25.0	34.5	3.7	375.7	457.5	3.5
269 Solrød	27.1	36.6	50.4	5.4	608.5	727.9	18.2
271 Vallø	12.9	17.3	23.8	2.6	170.8	227.4	2.7
025 Roskilde amtskommune	306.5	413.8	570.1	61.5	5715.3	7067.2	7.9

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
301 Bjergsted	13.2	8.7	14.8	1.8	186.9	225.3	1.6
303 Dianalund	12.4	8.2	14.0	1.7	118.8	155.0	2.3
305 Dragsholm	21.9	14.4	24.5	3.0	271.0	334.7	2.2
307 Fuglebjerg	11.0	7.3	12.3	1.5	180.3	212.4	1.5
309 Gørlev	10.4	6.9	11.6	1.4	128.2	158.5	1.7
311 Hashøj	11.2	7.2	12.3	1.5	203.2	235.5	1.8
313 Haslev	24.0	15.8	26.8	3.2	282.9	352.7	2.7
315 Holbæk	53.9	35.6	60.5	7.3	676.9	834.2	5.2
317 Hvidebæk	9.4	6.2	10.6	1.3	125.8	153.3	1.6
319 Høng	14.0	9.2	15.7	1.9	171.2	212.0	1.5
321 Jernløse	9.6	6.3	10.7	1.3	164.4	192.3	1.9
323 Kalundborg	33.8	22.3	37.8	4.6	213.3	311.8	2.4
325 Korsør	35.8	23.7	40.2	4.9	381.4	486.0	6.5
327 Nykøbing-Rørvig	31.9	7.9	13.4	1.6	95.6	130.3	3.3
329 Ringsted	49.7	32.8	55.7	6.7	811.5	956.5	3.2
331 Skælskør	19.2	12.7	21.6	2.6	206.8	262.9	1.5
335 Sorø	24.7	16.4	27.8	3.4	467.2	539.5	3.6
337 Stenlille	8.8	5.8	9.8	1.2	125.0	150.6	1.6
339 Svinninge	10.9	7.2	12.2	1.5	159.0	190.7	2.2
341 Tornved	15.4	10.1	17.2	2.1	194.8	239.6	2.3
343 Trundholm	18.2	12.0	20.4	2.5	320.2	373.2	2.3
345 Tølløse	15.9	10.6	17.9	2.2	243.2	289.8	2.3
030 Vestsjællands amtskommune	494.2	326.6	554.2	66.9	6416.5	7858.5	2.6

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
351 Fakse	21.5	15.4	21.8	6.9	192.4	258.0	1.8
353 Fladså	13.7	9.7	13.8	4.4	294.8	336.5	2.5
355 Holeby	8.7	6.2	8.8	2.8	70.6	97.2	0.8
357 Holmegård	32.4	8.8	12.5	4.0	114.0	151.8	2.3
359 Højreby	8.4	6.0	8.5	2.7	135.5	161.0	1.3
361 Langebæk	11.3	8.0	11.4	3.7	140.0	174.5	1.7
363 Maribo	21.9	15.5	22.1	7.0	303.7	370.3	2.4
365 Møn	21.3	15.1	21.4	6.8	234.6	299.2	1.3
367 Nakskov	30.8	21.8	31.1	9.9	142.3	235.9	7.2
369 Nykøbing Falster	47.7	33.9	48.2	15.4	427.2	572.2	4.3
371 Nysted	11.0	7.7	10.9	3.5	111.2	144.3	1.0
373 Næstved	85.7	60.8	86.4	27.6	723.7	984.2	4.9
375 Nr. Alslev	18.4	13.1	18.6	5.9	392.6	448.6	2.5
377 Præstø	13.2	9.4	13.3	4.2	217.9	258.0	2.4
379 Ravnsborg	12.5	8.9	12.6	4.0	91.0	129.0	0.7
381 Rudbjerg	7.3	5.2	7.4	2.4	60.6	82.8	0.6
383 Rødby	13.6	9.6	13.7	4.4	157.2	198.6	1.7
385 Rønnede	12.2	8.7	12.3	3.9	439.1	476.2	3.8
387 Sakskøbing	17.7	12.5	17.8	5.7	331.1	384.8	2.2
389 Stevns	20.0	14.2	20.2	6.4	139.8	200.6	1.2
391 Stubbekøbing	13.1	9.3	13.2	4.2	121.4	161.2	1.0
393 Suså	15.1	10.7	15.2	4.8	169.3	215.2	1.5
395 Sydfalster	12.7	9.1	12.9	4.1	147.2	185.9	1.6
397 Vordingborg	37.9	26.9	38.2	12.2	498.0	613.1	3.5
035 Storstrøms amtskommune	488.1	346.2	492.3	157.0	5655.3	7139.0	2.1

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1}) Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	Emission density tonne NO_2 km^{-2} a^{-1}
401 Allinge-Gudhjem	22.4	19.5	13.4	5.2	150.9	211.5	1.4
403 Hasle	18.4	16.0	11.0	4.3	137.9	187.5	1.6
405 Neksø	24.7	21.6	14.9	5.8	153.1	220.1	2.1
407 Rønne	41.6	36.4	25.0	9.7	195.3	308.0	10.6
409 Åkirkeby	19.0	16.5	11.3	4.4	192.8	243.9	1.3
411 Christiansø	0.3	0.0	0.3	0.0	0.9	1.2	3.0
040 Bornholms amtskommune	126.5	110.0	75.7	29.3	830.8	1172.2	2.0

No. Municipality	Emission (tonne NO_2 a^{-1}) Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	Emission density tonne NO_2 km^{-2} a^{-1}
421 Assens	12.3	11.9	26.7	2.0	191.9	244.8	1.8
423 Bogense	7.0	6.8	15.2	1.1	104.3	134.4	1.3
425 Broby	7.4	7.1	15.9	1.2	123.3	154.9	1.5
427 Egebjerg	9.8	9.5	21.4	1.6	202.7	245.1	2.0
429 Ejby	11.3	10.9	24.5	1.8	396.4	444.9	2.7
431 Fåborg	20.0	19.2	43.2	3.3	337.0	422.7	1.9
433 Glamsbjerg	6.7	6.4	14.4	1.1	127.6	156.2	1.7
435 Gudme	7.0	6.7	15.1	1.1	97.2	127.2	1.1
437 Hårby	5.7	5.5	12.3	0.9	86.1	110.6	1.4
439 Kerteminde	11.9	11.4	25.6	1.9	187.7	238.5	1.7
441 Langeskov	6.8	6.6	14.9	1.1	176.1	205.5	4.7
443 Marstal	4.2	4.1	9.1	0.7	35.1	53.1	3.2
445 Middelfart	21.1	20.4	45.9	3.5	425.0	515.9	7.1
447 Munkebo	6.6	6.5	14.5	1.1	89.5	118.2	6.2
449 Nyborg	21.0	20.3	45.6	3.4	365.3	455.7	5.5
451 Nr. Åby	6.1	5.8	13.1	1.0	234.1	260.2	4.0
461 Odense	199.5	192.8	433.0	32.6	2942.6	3800.5	12.5
471 Otterup	12.7	12.2	27.5	2.1	182.1	236.5	1.4
473 Ringe	12.7	12.3	27.6	2.1	297.0	351.6	2.3
475 Rudkøbing	7.9	7.7	17.2	1.3	102.9	136.9	2.2
477 Ryslinge	8.1	7.8	17.5	1.3	170.5	205.1	2.5
479 Svendborg	36.6	45.0	101.2	7.6	531.6	732.1	4.2
481 Sydlangeland	5.5	5.3	12.0	0.9	100.8	124.5	1.0
483 Søndersø	12.8	12.3	27.6	2.1	204.5	259.2	1.4
485 Tommerup	8.4	8.1	18.2	1.4	154.4	190.5	2.6
487 Tranekær	4.4	4.3	9.6	0.7	89.4	108.4	1.0
489 Ullerslev	5.5	5.4	12.0	0.9	137.0	160.8	3.0
491 Vissenbjerg	6.8	6.6	14.7	1.1	345.9	375.1	7.9
493 Ærøskøbing	5.1	4.9	11.0	0.8	72.9	94.8	1.3
495 Ørbæk	7.6	7.3	16.3	1.2	169.0	201.3	1.5
497 Årslev	10.5	10.0	22.5	1.7	169.1	213.8	2.9
499 Årup	6.0	5.8	13.0	1.0	193.1	218.8	2.7
042 Fyns amtskommune	524.8	506.8	1138.4	85.8	9042.1	11298.0	3.2

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
501 Augustenborg	10.1	14.5	17.2	1.8	126.2	169.7	3.2
503 Bov	16.1	22.9	27.3	2.8	534.0	603.1	4.1
505 Bredebro	6.1	8.7	10.3	1.1	91.2	117.4	0.8
507 Broager	9.5	13.5	16.0	1.6	134.5	175.2	4.0
509 Christiansfeld	13.8	19.8	23.6	2.4	396.8	456.5	2.2
511 Gram	8.0	11.3	13.5	1.4	131.5	165.7	1.3
513 Gråsten	10.6	15.2	18.1	1.8	170.6	216.4	3.8
515 Haderslev	46.6	66.5	79.1	8.1	605.9	806.2	2.9
517 Højer	4.7	6.7	8.0	0.8	69.9	90.0	0.8
519 Lundtoft	9.5	13.6	16.2	1.7	308.1	349.1	2.5
521 Løgumkloster	10.7	15.3	18.2	1.9	179.5	225.6	1.1
523 Nordborg	22.9	32.7	38.8	4.0	233.8	332.1	2.7
525 Nr. Rangstrup	15.4	22.0	26.1	2.7	309.3	375.5	1.2
527 Rødding	16.8	24.1	28.6	2.9	301.4	373.8	1.4
529 Rødekro	17.0	24.3	28.8	2.9	476.5	549.5	2.7
531 Skærbæk	11.9	17.0	20.2	2.1	260.5	311.7	0.9
533 Sundeved	8.2	11.6	13.8	1.4	156.0	191.0	2.8
535 Sydals	10.3	14.7	17.5	1.8	121.5	165.9	1.8
537 Sønderborg	42.9	61.2	72.8	7.4	423.3	607.6	11.2
539 Tinglev	16.1	23.0	27.3	2.8	267.4	336.5	1.0
541 Tønder	19.4	27.8	33.0	3.4	321.7	405.2	2.2
543 Vojens	26.1	37.2	44.2	4.5	583.3	695.3	2.3
545 Åbenrå	32.9	47.1	55.9	5.7	582.8	724.4	5.6
050 Sønderjyllands amtskommune	385.6	550.6	654.5	66.9	6785.7	8443.3	2.1

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
551 Billund	7.7	9.6	20.1	1.9	145.2	184.5	1.2
553 Blåbjerg	6.7	B.3	17.5	1.6	177.4	211.6	0.8
555 Blåvandshuk	4.0	5.0	10.6	1.0	102.1	122.7	0.6
557 Bramming	13.0	16.3	34.2	3.2	255.7	322.5	1.9
559 Brørup	6.3	7.9	16.5	1.5	138.7	170.9	1.6
561 Esbjerg	82.7	103.6	217.2	20.1	1155.7	1579.2	7.2
563 Fanø	3.3	4.1	8.6	0.8	25.1	41.8	0.7
565 Grindsted	17.4	21.8	45.7	4.2	419.7	508.9	1.3
567 Helle	8.5	10.5	22.1	2.0	206.2	249.4	0.9
569 Holsted	7.0	8.7	18.2	1.7	187.1	222.7	1.2
571 Ribe	18.3	22.9	47.9	4.4	446.8	540.3	1.5
573 Varde	19.2	24.0	50.3	4.7	428.6	526.8	2.1
575 Vejen	16.3	20.7	42.9	4.0	404.3	488.2	2.0
577 Ølgod	11.4	14.3	30.0	2.8	209.1	267.5	1.1
055 Ribe amtskommune	221.6	277.7	581.9	53.9	4301.6	5436.8	1.7

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
601 Brædstrup	12.1	26.3	28.4	3.8	202.1	272.8	1.4
603 Børkop	15.3	33.1	35.8	4.8	363.9	453.0	4.4
605 Egtved	20.3	43.6	47.2	6.4	423.5	541.0	1.7
607 Fredericia	66.2	142.7	154.6	20.9	836.2	1220.6	9.1
609 Gedved	13.8	29.9	32.4	4.4	253.3	333.8	2.2
611 Give	19.6	42.2	45.7	6.2	370.4	484.1	1.2
613 Hedensted	20.8	44.9	48.6	6.6	341.1	462.0	3.4
615 Horsens	78.8	169.9	184.0	24.9	888.1	1345.7	7.1
617 Jelling	7.1	15.4	16.7	2.3	89.2	130.7	1.5
619 Juelsminde	21.2	45.6	49.4	6.7	295.3	418.2	1.7
621 Kolding	82.2	177.1	191.8	25.9	1309.0	1786.0	7.5
623 Lunderskov	6.9	14.8	16.1	2.2	113.4	153.4	1.6
625 Nr. Snede	10.5	22.7	24.6	3.3	331.8	392.9	1.5
627 Tørring-Uldum	16.7	35.8	38.8	5.2	458.8	555.3	2.9
629 Vamdrup	9.8	21.1	22.9	3.1	168.8	225.7	2.2
631 Vejle	73.1	157.8	170.9	23.1	1113.3	1538.2	10.7
060 Vejle amtskommune	474.5	1022.9	1107.9	149.8	7558.2	10313.3	3.4

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
651 Avlum-Haderup	9.0	64.2	22.8	1.7	193.4	291.0	1.2
653 Brande	11.4	83.1	29.4	2.2	176.6	302.8	1.6
655 Egvad	12.9	93.7	33.1	2.5	299.0	441.2	1.2
657 Herning	75.4	548.6	193.9	14.5	1297.2	2129.6	3.9
659 Holmsland	7.1	51.7	18.3	1.4	205.5	284.0	3.0
661 Holstebro	51.3	373.5	132.0	9.9	714.1	1280.9	3.6
663 Ikast	29.3	213.4	75.4	5.7	538.3	862.1	2.9
665 Lemvig	25.7	187.4	66.3	5.0	331.5	615.9	1.3
667 Ringkøbing	22.8	166.1	58.7	4.4	373.6	625.6	1.6
669 Skjern	16.9	123.0	43.5	3.3	249.1	435.7	1.3
671 Struer	25.5	186.0	65.8	4.9	314.1	596.3	3.4
673 Thyborøn-Harboør	7.1	51.6	18.2	1.4	91.3	169.6	4.0
675 Thyholm	5.2	37.2	13.3	1.0	103.8	160.5	2.1
677 Trehøje	12.4	90.0	31.8	2.4	165.4	301.9	1.0
679 Ulfborg-Vemb	9.5	69.1	24.4	1.8	140.6	245.5	1.1
681 Videbæk	16.0	116.8	41.3	3.1	266.8	444.0	1.5
683 Vinderup	11.0	79.0	28.1	2.1	241.1	361.3	1.6
685 Åskov	9.1	66.4	23.5	1.8	190.8	291.6	1.2
065 Ringkøbing amtskommune	357.8	2600.6	919.8	69.0	5892.4	9839.5	2.0

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
701 Ebeltoft	13.6	55.1	30.1	3.3	290.8	392.9	1.4
703 Galten	10.4	42.1	23.0	2.5	177.4	255.5	3.5
705 Gjern	7.7	31.3	17.1	1.9	195.2	253.1	1.8
707 Grenå	19.8	80.3	43.9	4.8	265.6	414.4	2.1
709 Hadsten	11.5	46.6	25.4	2.8	313.7	399.9	2.9
711 Hammel	10.5	42.7	23.3	2.6	268.3	347.5	2.4
713 Hinnerup	10.5	42.9	23.4	2.6	159.2	238.5	3.1
715 Hørning	7.9	32.2	17.6	1.9	219.7	279.4	4.1
717 Langå	8.8	35.4	19.5	2.2	161.7	227.6	1.7
719 Mariager	8.7	35.2	19.2	2.1	163.2	228.4	1.1
721 Midtdjurs	8.0	32.4	17.7	1.9	156.3	216.3	1.2
723 Nørhald	9.3	37.7	20.6	2.3	170.8	240.6	1.2
725 Nr. Djurs	8.2	33.3	18.2	2.0	144.1	205.7	0.9
727 Odder	19.5	79.3	43.3	4.7	300.5	447.4	2.0
729 Purhus	9.5	38.1	20.9	2.3	209.4	280.2	1.7
731 Randers	65.2	265.0	144.7	15.8	1094.5	1585.2	10.3
733 Rosenholm	10.1	40.9	22.3	2.4	291.8	367.6	2.6
735 Rougsø	8.6	34.9	19.1	2.1	121.7	186.4	0.8
737 Ry	10.3	42.0	22.9	2.5	242.9	320.6	2.1
739 Rønde	6.3	25.7	14.0	1.5	234.7	282.2	2.8
741 Samsø	4.7	19.2	10.5	1.1	59.9	95.5	0.8
743 Silkeborg	51.1	208.0	113.6	12.4	785.6	1170.9	4.6
745 Skanderborg	21.1	85.3	46.6	5.1	624.3	782.3	5.5
747 Sønderhald	8.7	35.3	19.3	2.1	280.8	346.2	2.5
749 Them	6.8	26.9	14.7	1.6	228.9	278.8	1.3
751 Århus	275.2	1118.5	610.9	66.9	3762.9	5834.4	12.4
070 Århus amtskommune	632.0	2566.3	1401.9	153.5	10924.1	15677.8	3.4

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
761 Bjerringbro	22.0	40.1	32.0	5.1	278.1	377.2	1.8
763 Fjends	13.8	25.6	20.1	3.2	194.7	257.3	1.1
765 Hanstholm	10.2	18.4	14.7	2.3	115.4	161.0	0.7
767 Hvorslev	10.9	20.4	16.0	2.5	101.1	150.9	1.2
769 Karup	11.4	21.4	16.7	2.6	183.5	235.6	1.4
771 Kjellerup	22.6	41.3	33.0	5.2	431.1	533.2	2.1
773 Morsø	40.7	74.1	59.1	9.4	387.7	571.0	1.6
775 Møldrup	12.5	22.6	18.0	3.3	230.6	287.0	1.4
777 Sallingsund	10.7	19.4	15.5	2.5	149.4	197.4	2.0
779 Skive	45.5	82.7	66.0	10.5	486.2	691.0	3.0
781 Spøttrup	13.4	24.4	19.4	3.1	174.3	234.6	1.2
783 Sundsøre	11.7	21.3	17.0	2.7	105.4	158.2	0.9
785 Sydthy	21.0	38.8	30.5	4.8	250.5	345.6	1.1
787 Thisted	50.4	91.7	73.2	11.6	664.0	890.9	1.6
789 Tjele	14.0	25.4	20.3	3.2	285.4	348.3	1.3
791 Viborg	67.0	121.9	97.3	15.5	831.8	1133.5	3.6
793 Ålestrup	12.8	23.3	18.6	3.4	194.2	252.2	1.4
076 Viborg amtskommune	390.6	712.6	567.5	91.0	5063.4	6825.1	1.7

Table 2 continued

No. Municipality	Emission (tonne NO_2 a^{-1})						Emission density
	Domestic heating	Energy generation	Industrial combustion	Industrial production	Road traffic	Total	tonne NO_2 km^{-2} a^{-1}
801 Arden	10.8	21.5	17.5	39.8	278.0	367.5	1.6
803 Brovst	11.4	22.7	18.5	42.1	164.3	259.0	1.2
805 Brønderslev	26.8	53.6	43.6	99.0	398.9	621.9	2.0
807 Dronninglund	20.2	40.4	32.8	74.6	417.6	585.6	1.9
809 Farsø	10.6	21.1	17.2	39.1	173.7	261.7	1.3
811 Fjerritslev	11.0	21.9	17.8	40.5	211.7	302.9	1.0
813 Frederikshavn	47.0	94.0	76.5	173.9	564.9	956.3	5.3
815 Hadsund	13.8	27.7	22.5	51.2	220.9	336.2	2.0
817 Hals	14.2	28.3	23.0	52.3	172.5	290.2	1.5
819 Hirtshals	19.2	38.4	31.3	71.1	268.9	428.9	2.2
821 Hjørring	45.5	91.1	74.1	168.4	623.2	1002.3	3.2
823 Hobro	18.5	37.2	30.2	67.9	387.7	541.6	3.3
825 Læsø	3.4	6.8	5.5	12.5	43.7	71.9	0.6
827 Løgstør	14.2	28.4	23.1	52.5	202.6	320.7	1.5
829 Løkken-Vrå	11.7	23.5	19.1	43.4	176.1	273.7	1.5
831 Nibe	9.9	19.7	16.0	36.4	127.0	209.0	1.1
833 Norager	7.3	14.6	11.9	26.6	193.1	253.5	1.5
835 Pandrup	13.8	27.5	22.4	50.9	234.1	348.7	1.8
837 Sejlflod	12.0	23.8	19.4	44.0	137.6	236.8	1.1
839 Sindal	12.7	25.4	20.7	47.1	206.5	312.4	1.3
841 Skagen	18.4	36.8	29.9	68.0	248.0	401.0	2.8
843 Skørping	12.6	25.2	20.5	46.6	197.9	302.8	1.3
845 Støvring	15.8	31.6	25.7	58.4	370.9	502.5	2.3
847 Sæby	24.0	47.9	39.0	88.7	442.3	641.9	2.0
849 Åbybro	14.6	29.1	23.7	53.8	245.9	367.2	2.1
851 Ålborg	204.3	408.8	332.5	756.0	2626.4	4328.0	7.7
861 Års	16.5	33.0	26.8	61.0	230.1	367.5	1.6
080 Nordjyllands amtskommune	640.0	1280.0	1041.0	2365.8	9564.6	14891.5	2.4

Table 3. Total NO_x emissions from large point sources for the counties in Denmark.

No. County	Emission in tonne NO_2 a^{-1}
013 København and Frederiksberg	13315.
020 Frederiksborg amtskommune	2300.
030 Vestsjællands amtskommune	42819.
035 Storstrøms amtskommune	1530.
042 Fyns amtskommune	33200.
050 Sønderjyllands amtskommune	11780.
055 Ribe amtskommune	9590.
060 Vejle amtskommune	6932.
070 Århus amtskommune	3069.
080 Nordjyllands amtskommune	6120.
Total	130655.

Table 4. Estimated NO_x emissions from maritime vessels.

	Tonne NO_2 a^{-1}
International freighter traffic	16678.
Main national ferry services	7652.
Total	24330.

Table 5. Total NO_x emissions.

Source	Tonne NO_2 a^{-1}	%
Domestic heating	7438.	2.53
Energy generation	12892.	4.39
Combustion in industry	12736.	4.33
Industrial production	4029.	1.37
Road traffic	101775.	34.63
Maritime vessels	24331.	8.28
Large point sources	130655.	44.46
Total	293856.	99.99

The difference between the summary of »road traffic« in the Tables 1 and 2, and the figure given here in Table 5, arise from rounding off and from the gridding of the emissions from roads. Where some km^2 with emissions, do not exist in the land-use registration and therefore cannot be assigned to a municipality. This is e.g. caused by the fact that there is traffic on bridges between islands and these bridges are in the land-use survey registered as sea.

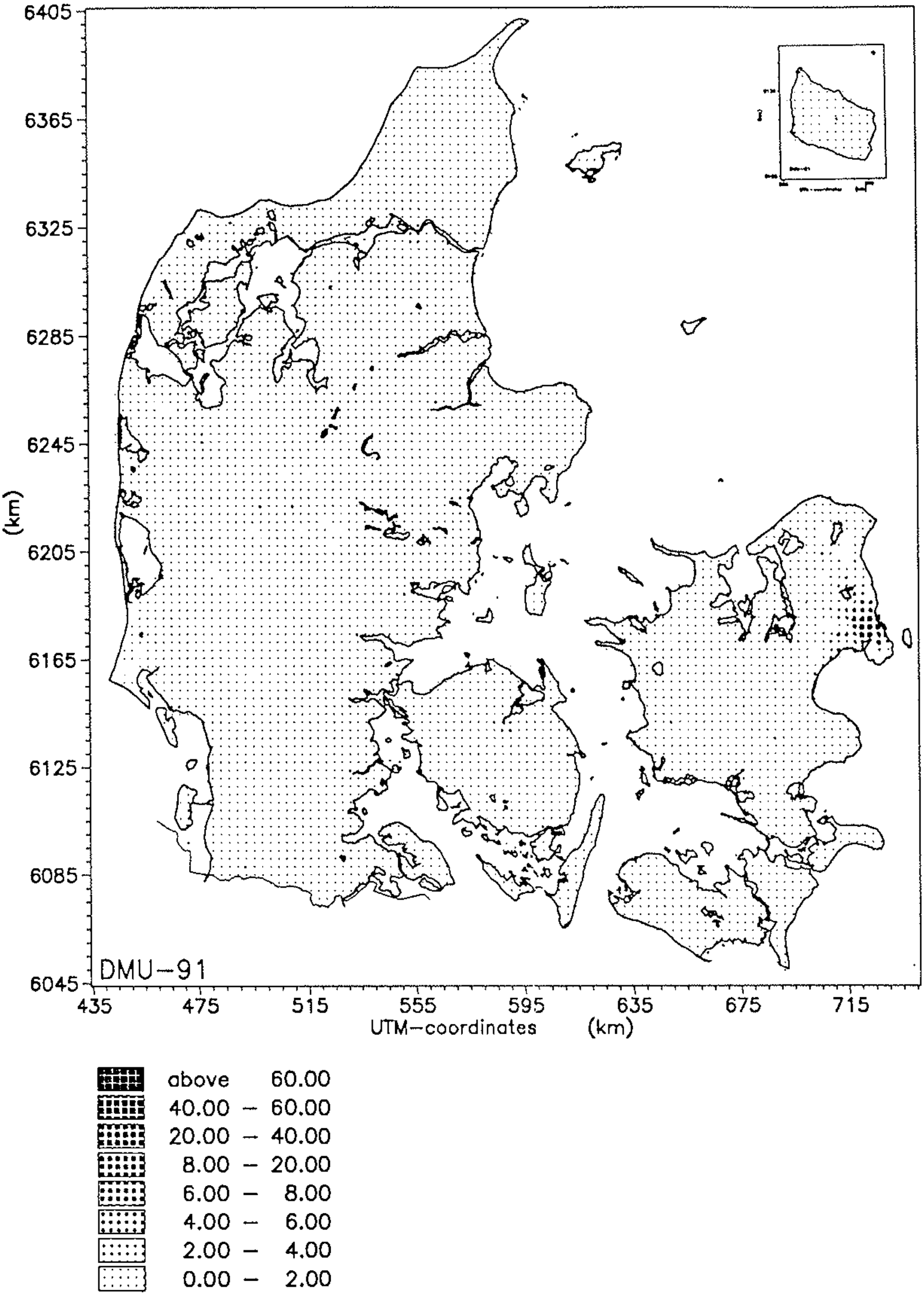

Figure 1. NO_x emissions density from domestic heating (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

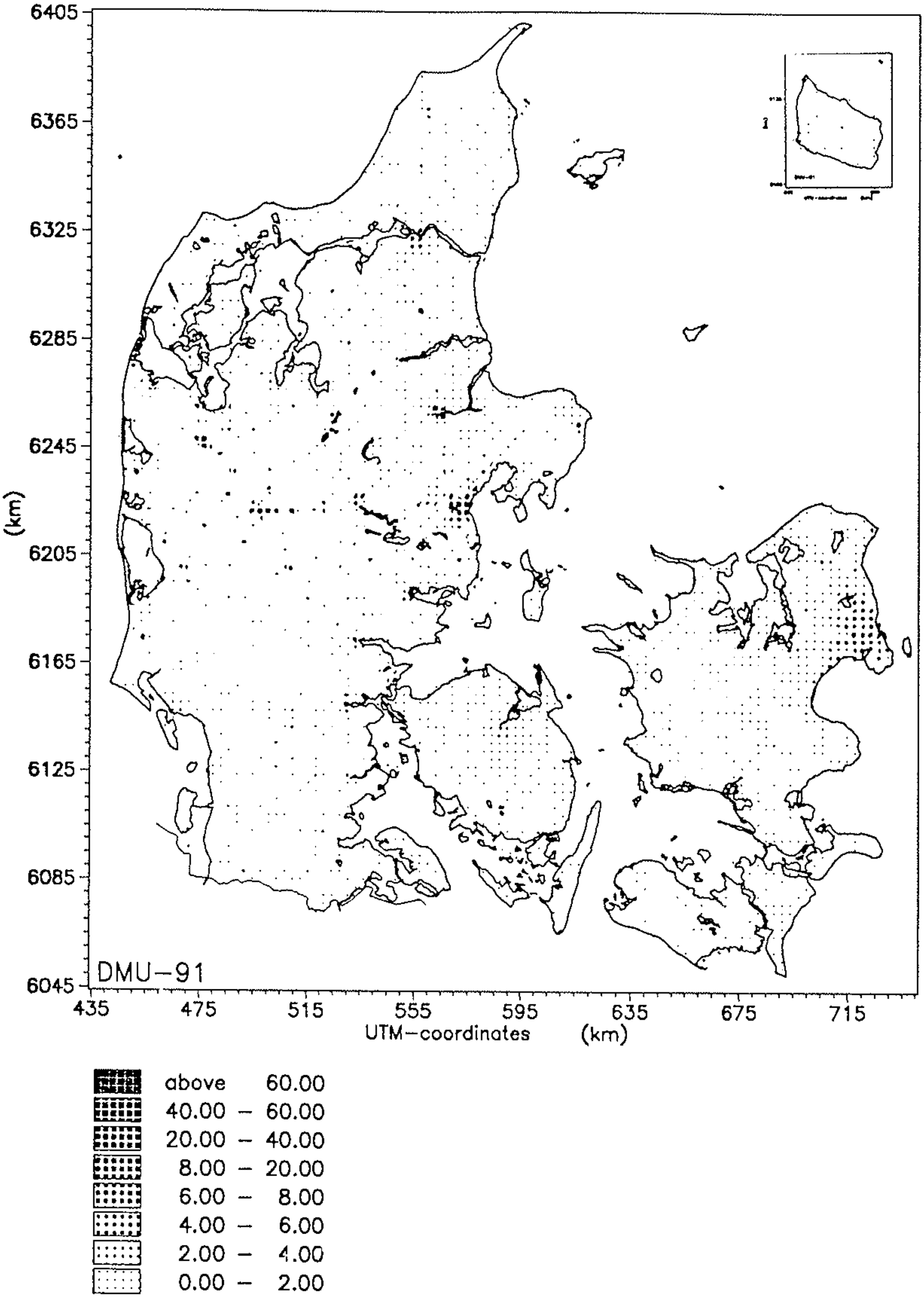

Figure 2. NO_x emissions density from energy generation (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

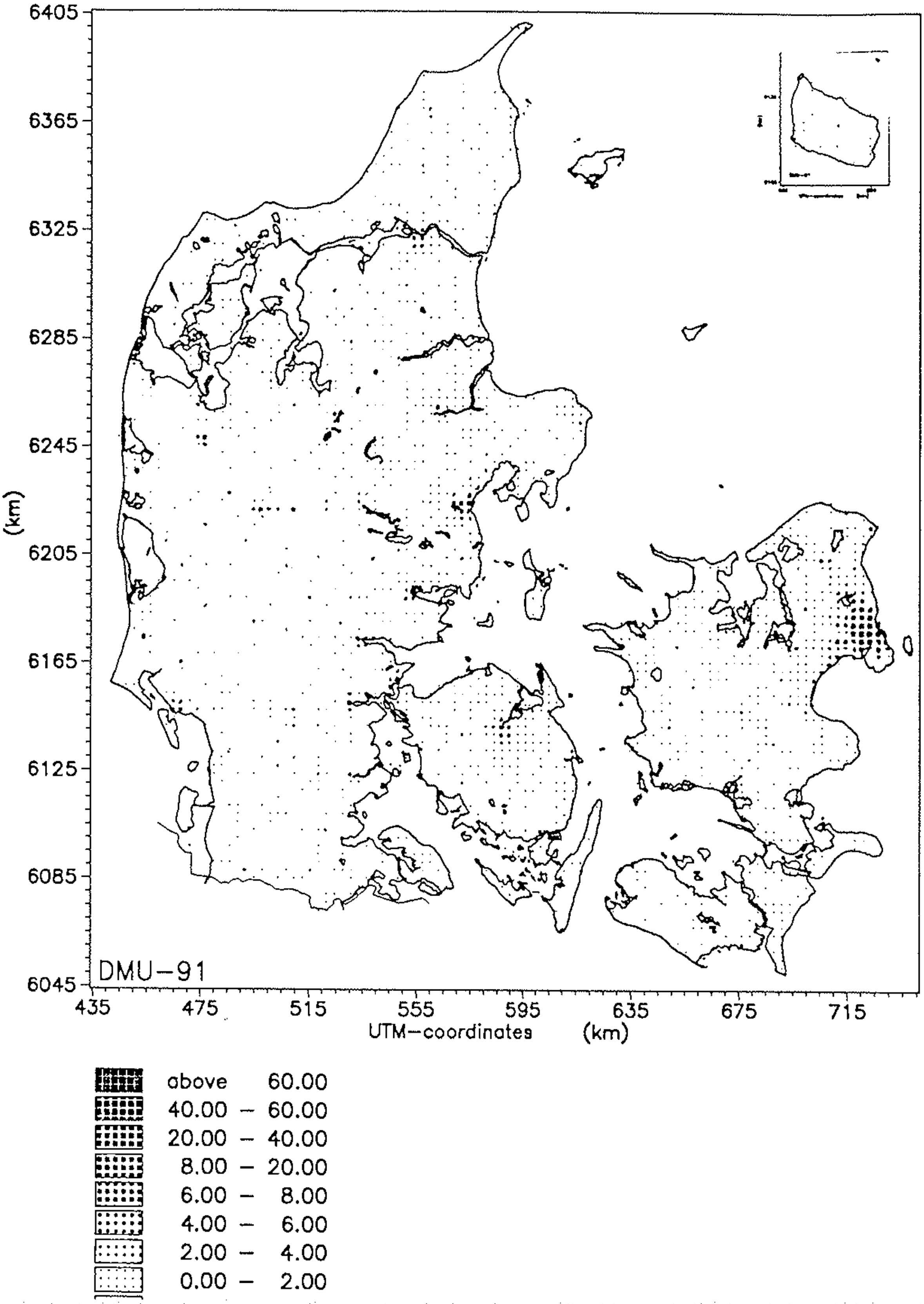

Figure 3. NO_x emissions density from industrial combustion (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

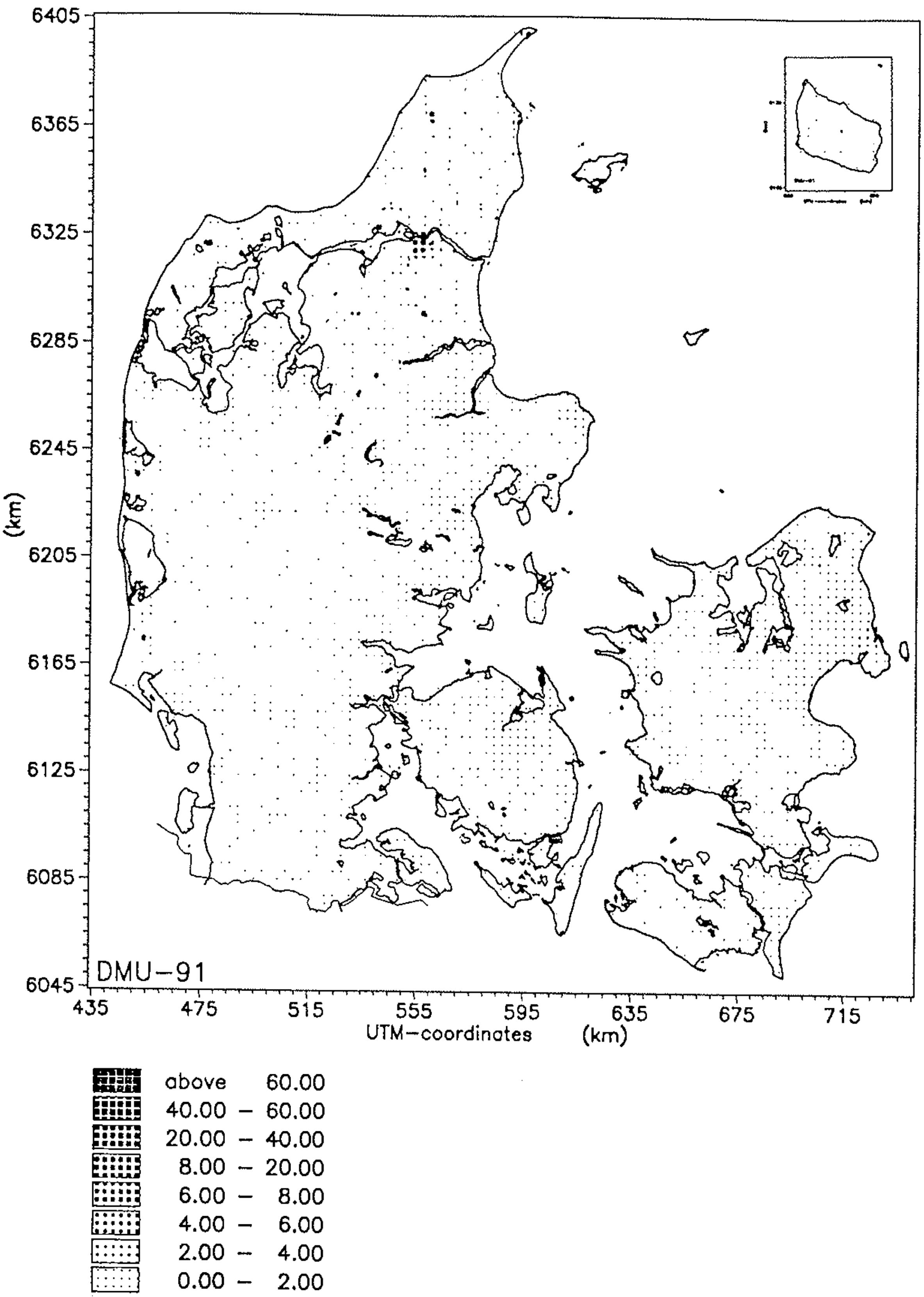

Figure 4. NO_x emissions density from industrial production (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

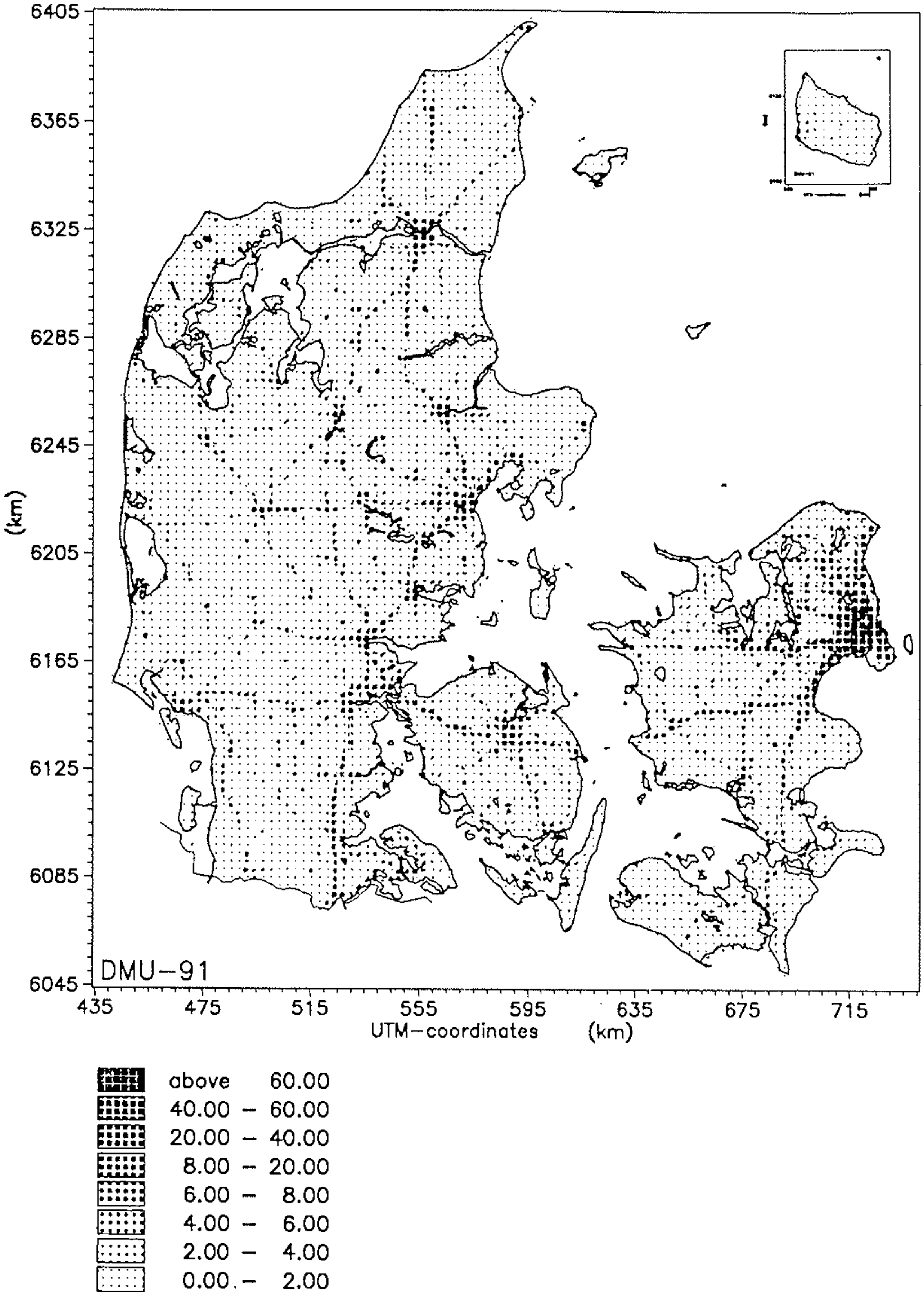

Figure 5. NO_x emissions density from road traffic (tonne NO_2 km^{-2} a^{-1}).
(Note that the scale is not linear).

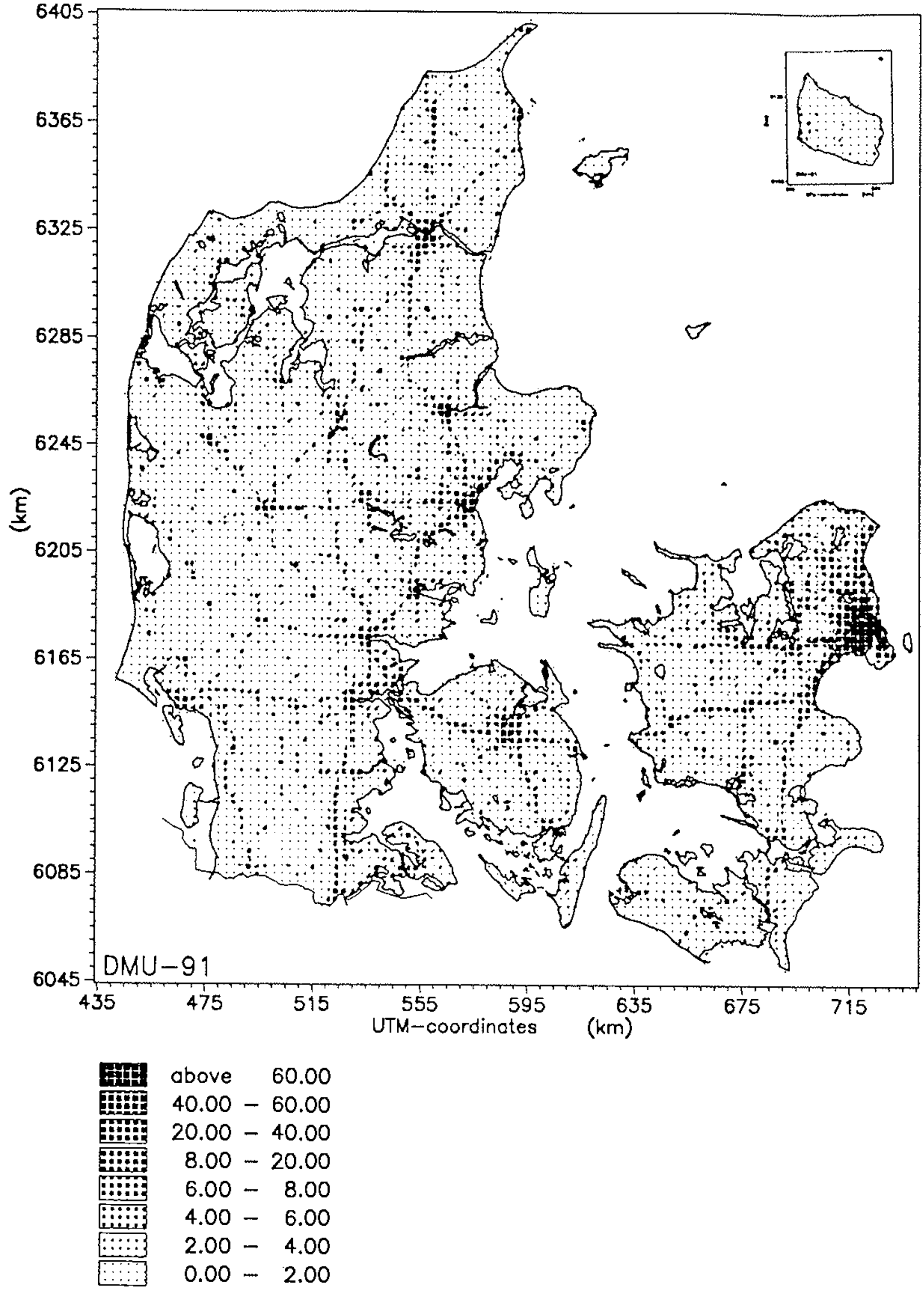

Figure 6. Total NO_x emissions density from area sources (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

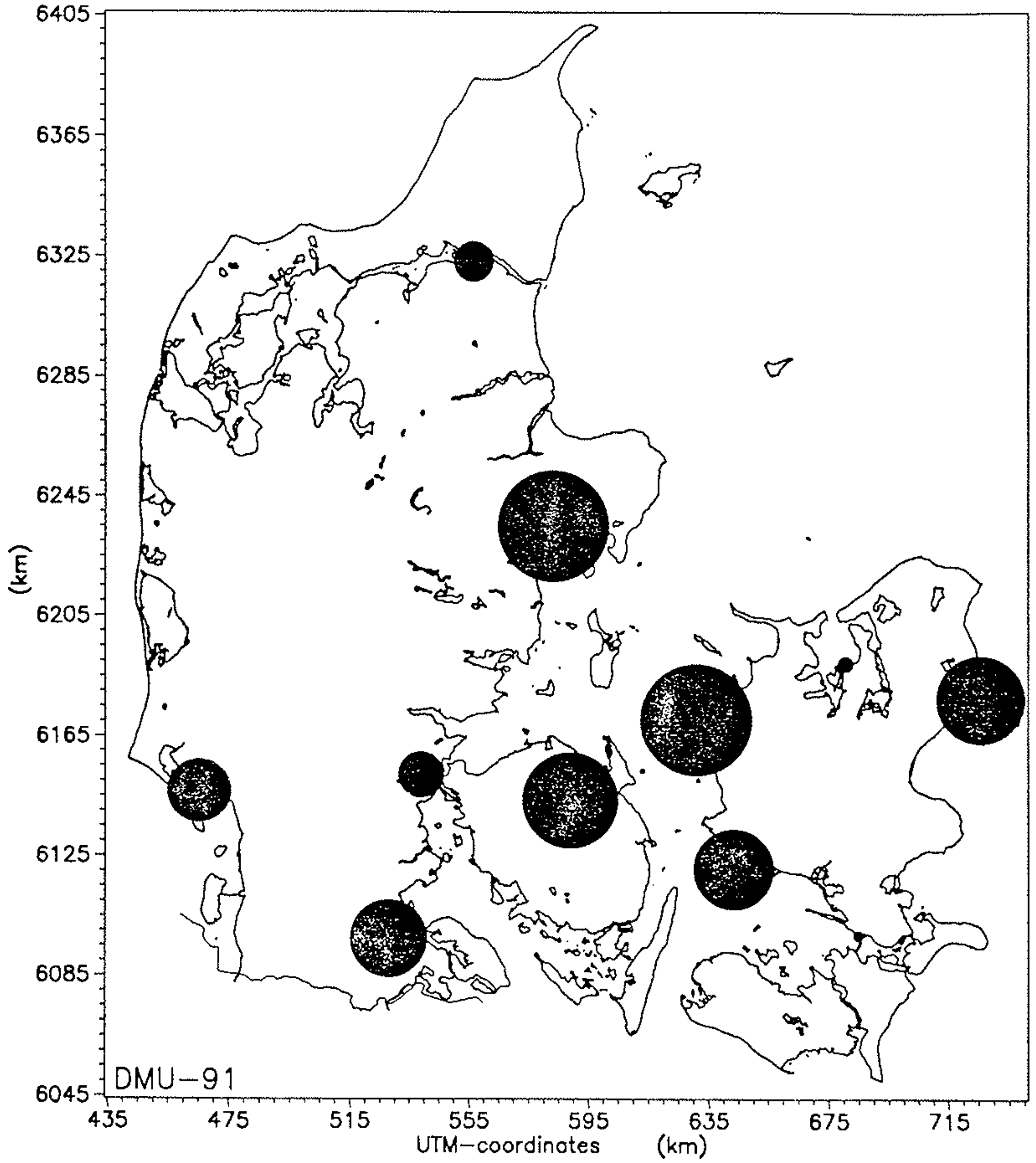

Figure 7. NO_x emissions from major Danish point sources. The diameter of a circle shows the relative magnitude of the source. Sources near each other have been clustered for clarity.

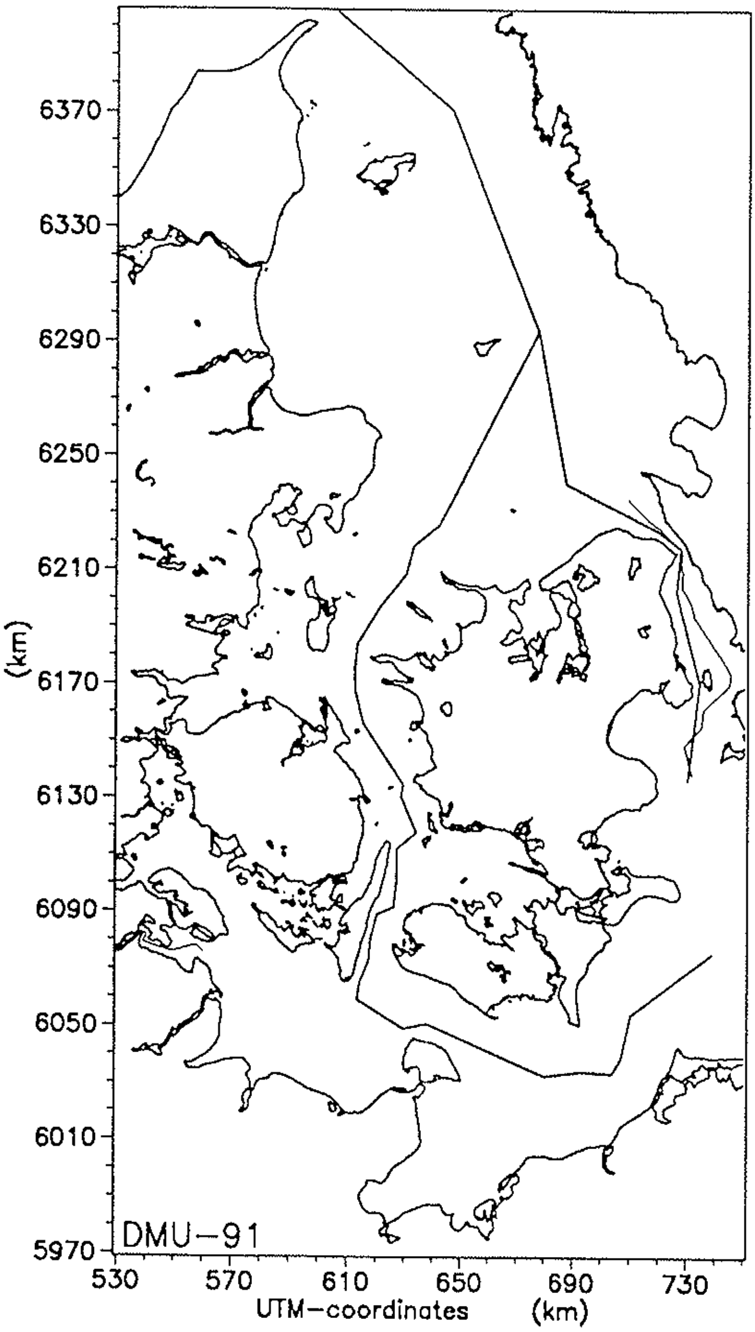

Figure 8. Main freighter routes through Danish waters.

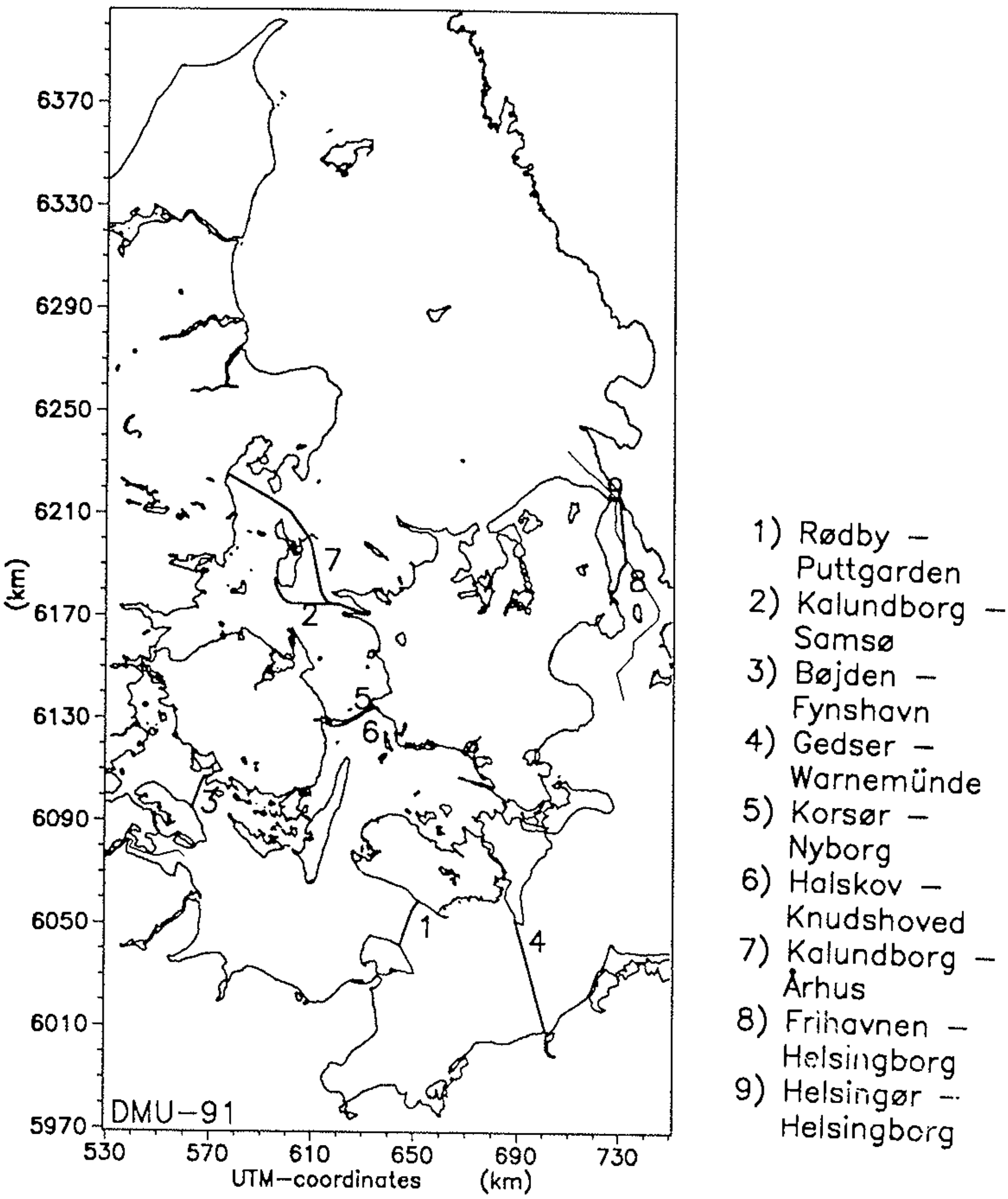

Figure 9. Main ferry services run by DSB.

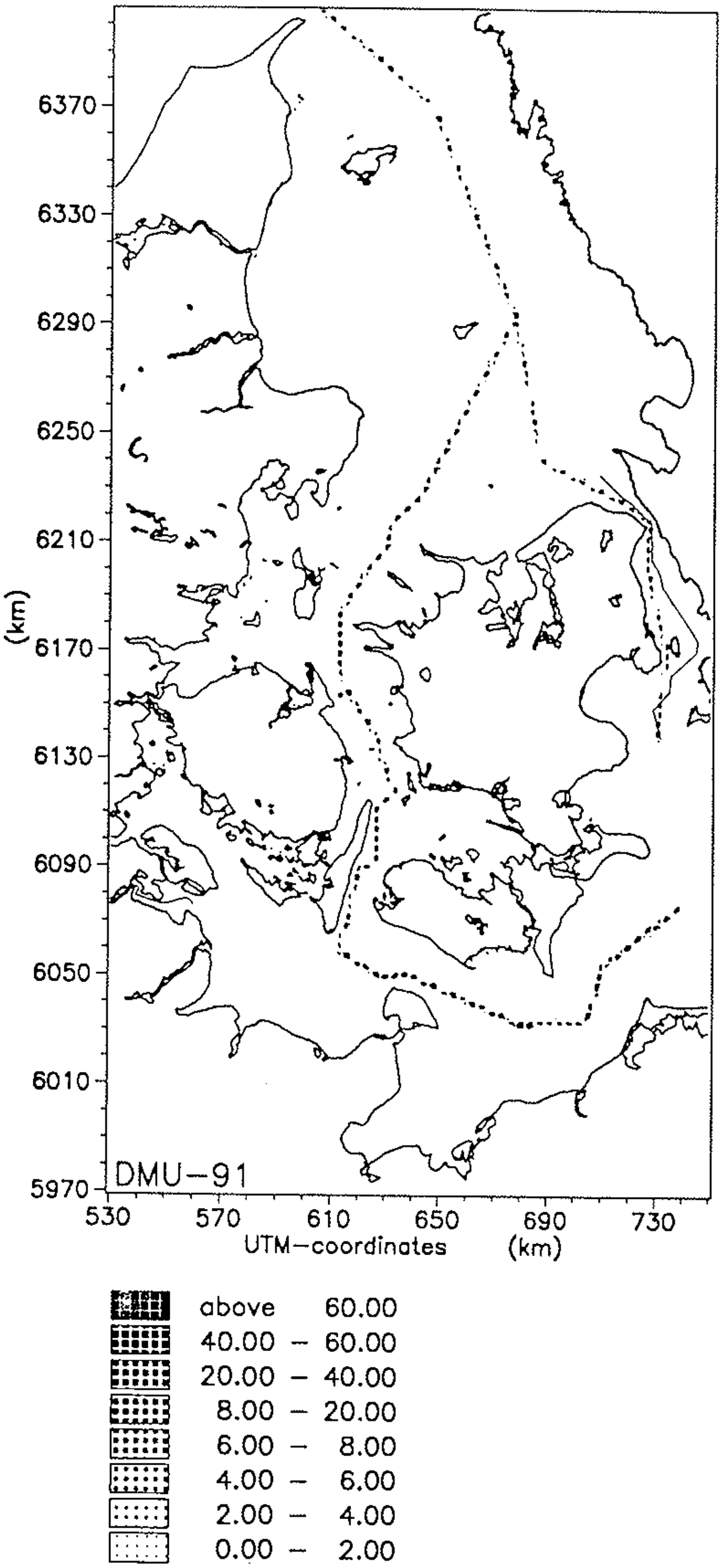

Figure 10. NO_x emissions density from international freight traffic (tonne NO_2 km^{-2} a^{-1}).
(Note that the scale is not linear).

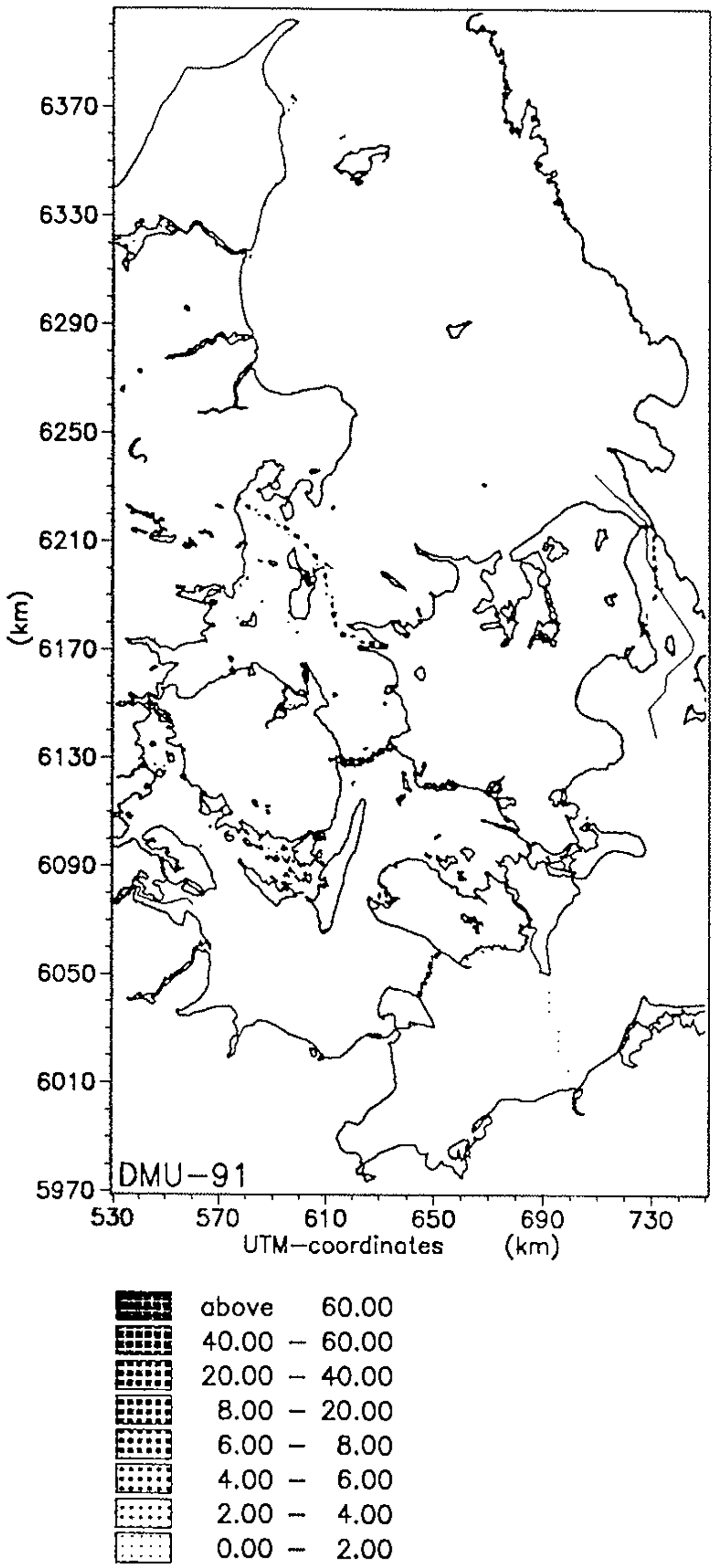

Figure 11. NO_x emissions density from main ferry services (tonne NO_2 km^{-2} a^{-1}). (Note that the scale is not linear).

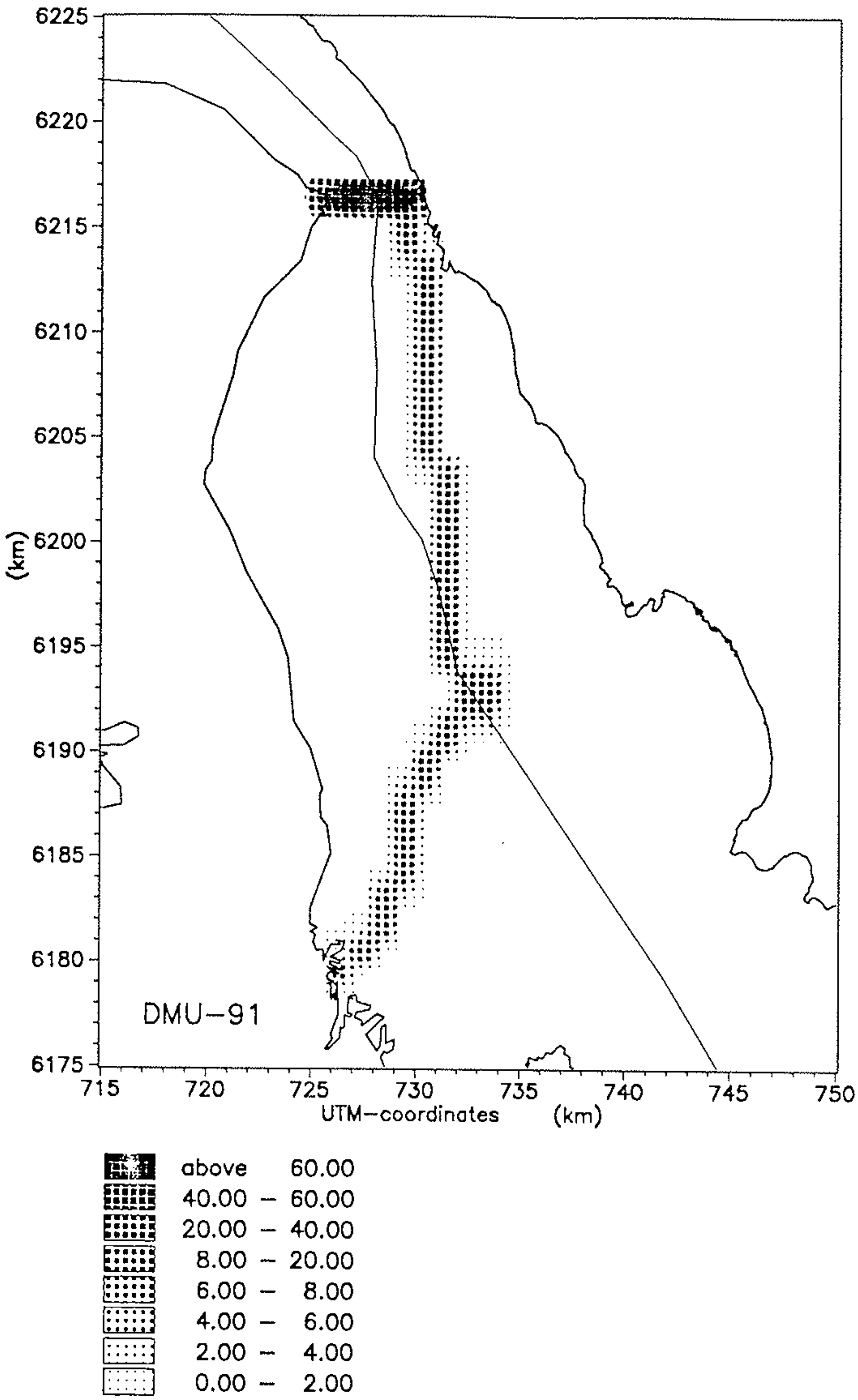

Figure 12. NO_x emissions density from ferry services within the Sound (tonne NO_2 km^{-2} a^{-1}).
(Note that the scale is not linear).

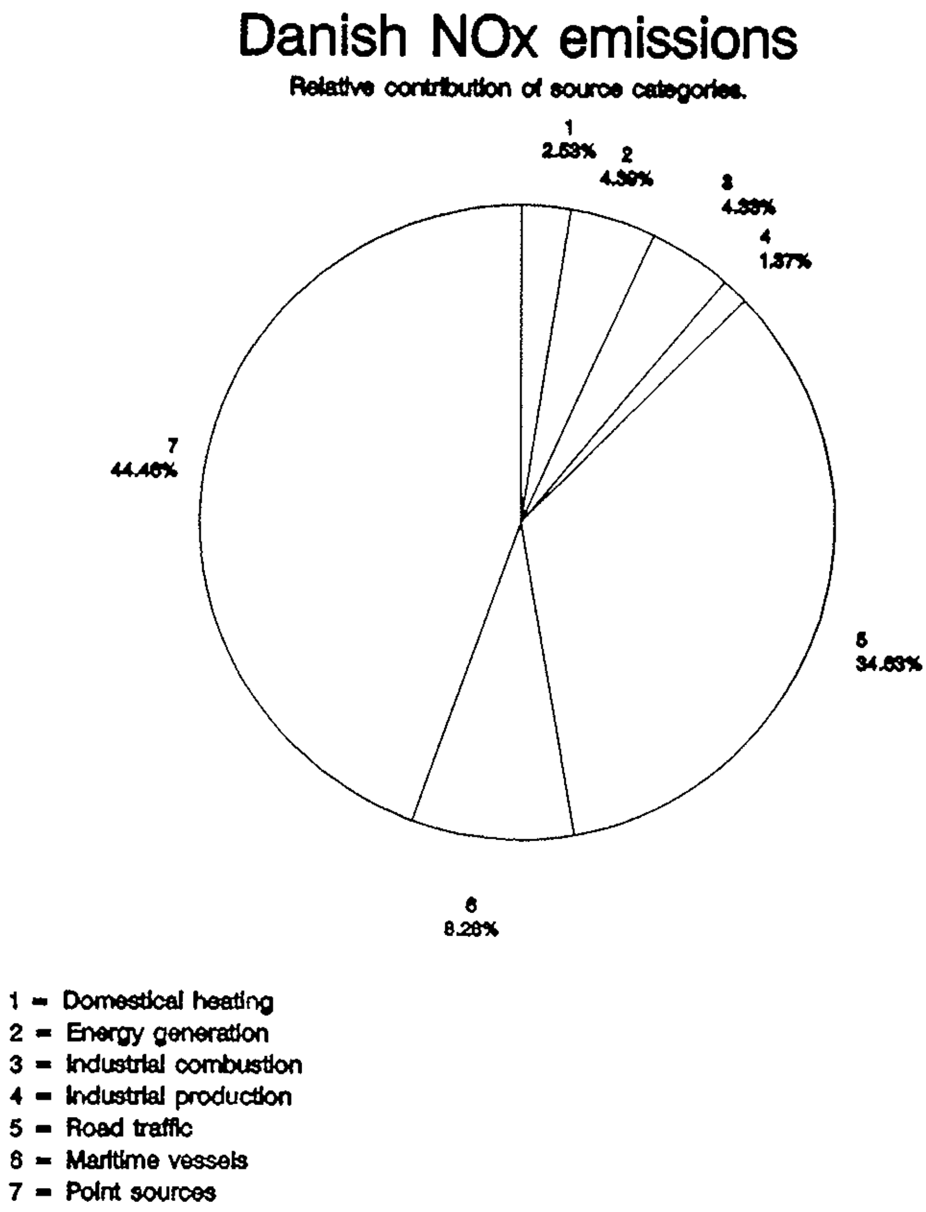

Figure 13. Relative contributions of the different source categories to the total NO_x emission.

Appendix A

Main computer programs used

```
************************************************************************
*                                                                      *
*   Fortran program ROAD_TRAFFIC                                       *
*   DMU-FOLU, 910117, Erik Runge.                                      *
*                                                                      *
*   Reads the data from the Road Data Laboratory. Data containing      *
*   information about Danish state- and county roads.                  *
*   OBS - on the tape from the Road Data Laboratory the seperation     *
*         mark between values, is ',' - but sometimes, at the          *
*         end of a line one or perhaps more ',' is missing.            *
*         Before this program is run, the missing ','s are             *
*         to be inserted                                               *
*   From the information in the road number and the county             *
*   number a 'flag' is set to indicate which of the procedures         *
*   in subroutine SYS34_UTM is to be used to transform the coordi-     *
*   nates from system 34 to UTM Zone 32. The coordinates which         *
*   in subroutine SYS34_UTM are tranformed to UTM Zone 33 coordi-      *
*   nates (Bornholm), are, from the subroutine SYS34_UTM, by call      *
*   of subroutine UTMUTM transformed to UTM Zone 32 coordinates.       *
*   OBS Coordinates where either X or Y for either begin or end        *
*       of a road piece are missing, are if the roadpiece is           *
*       less than 1 km assigned to a grid element, if the piece        *
*       is longer than 1 km the emission is distributed over the       *
*       county by the number of inhabitants.                           *
*   The NOx emission is calculated on basis of information from traf-  *
*   fic work and emission factors from Corinair                        *
*   The NOx emission is 'gridded' on a UTM Zone 32 grid.               *
*   Finally the NOx emission from municipal roads are calculated       *
*   and distributed over km2, after the number of inhabitants.         *
*                                                                      *
* The variables;                                                       *
*    Read from infile (file 10);                                       *
*        AMT     - The county (amt) to which the road piece            *
*                  belongs. If AMT = 0 it is a stateroad which         *
*                  does not belong to a specific county.               *
*        ROAD    - The number given to the road by the Road Data Lab.  *
*                  From the number one can tell where in the           *
*                  country the road is placed.                         *
*        EXTRA   - An extra integer, the meaning of which I            *
*                  do not know.                                        *
*        FROM    - The km-stone and m from the stone the road          *
*                  piece is measured from.                             *
*        TO      - The km-stone and m from the stone the road          *
*                  piece is mesaured to.                               *
*        DIST    - Length of the road piece.                           *
*        SPEED   - The max allowed speed on the road.                  *
*        TRAFY   - Average traffic in a day and night of the year.     *
*        TRAFJ   - Average traffic in a day ad night in the month      *
*                  of July.                                            *
*        WORK    - Traffic work in mio waggon km (trafikarbejdet).     *
*        FACL    - The look of the left side of the road.              *
*        FACR    - The look of the right side of the road;             *
*             1 - none or very dispersed ribbon buildings.             *
*             2 - ribbon buildings, with no fronts.                    *
*             3 - industry.                                            *
*             4 - houses and other low buildings.                      *
*             5 - habitation, buildings of flats.                      *
*             6 - shop fronts.                                         *
*        XBEG    - X coordinate of start point.                        *
*        XEND    - X coordinate of end   point.                        *
*        YBEG    - Y coordinate of start point.                        *
*        YEND    - Y coordinate of end   point.                        *
*                  Coordinates are read in system 34 transformed       *
*                  and written to file in UTM Zone 32.                 *
*    Arrays;                                                           *
*        A_REST - Holds the values of WORK from records where          *
```

```
*               coordinates are missing, but can be assigned       *
*               to a county                                        *
*        OUT    - Holds the calculated emission in each km2        *
*               Values written to OUT in subroutine GRID,          *
*               MUNIC_ROAD, REST                                   *
*                                                                  *
*    Auxilary variables;                                           *
*        EMI    - The calculated emission                          *
*        FLAG   - Determins which procedure in subroutine SYS34_UTM *
*               is to be used.                                     *
*        TWO    - Used when plotting data (Keep pen down).         *
*        THREE  - Used when plotting data (Pen up and down).       *
*        PIX    - Used when plotting data (Width of plotted line   *
*        I      - Counts the traverses (K) of the program.         *
*                                                                  *
* The files;                                                       *
*        File 10 is the file from which the data are read.         *
*        File 11 is the file to which the transformed data         *
*               are written.                                       *
*        File 12 is the file to which records are written if       *
*               coordinates are missing                            *
*        File 13 is the file to which error messages is written    *
*                                                                  *
*  Necessary files;                                                *
*        To run the program files must exist for each county and   *
*        for each municipality with information from the 'land     *
*        use' registration on which km2 is covered by the          *
*        county/municipality. These files are used in the sub-     *
*        routines REST and MUNIC_ROAD to distribute the emission   *
*                                                                  *
*  Subroutines called;                                             *
*        subroutine FAIL(F_FILE,TEXT,A,B)                          *
*        subroutine FLAG_ROAD(AMT,ROAD,FLAG)                       *
*        subroutine GRID(BEGX,BEGY,ENDX,ENDY,EMI)                  *
*        subroutine MUNIC_ROAD(W_SUM)                              *
*        subroutine REST(NR,WORK)                                  *
*        subroutine NOX_EMI_ROAD(WORK,EMI)                         *
*        subroutine SYS34_UTM(FLAG,X,Y)                            *
*        subroutine UTMUTM(X,Y,DIR)                                *
*                                                                  *
********************************************************************
c
      program ROAD_TRAFFIC
c
      integer AMT,DIST,EXTRA,FACL,FACR,FLAG,FROM,I,J,K,ROAD,
     1        SPEED,TO,TRAFJ,TRAFY
      double precision A_REST,BEGY,BEGX,EMI,ENDY,ENDX,OUT,
     1                 WORK,W_SUM,SUM,EMISUM

      character*3 NR
      common A_REST(1:14),OUT(435:900,6045:6405),EMISUM
c
      open(10,file='MSLUFT$MET:[HAVDATA.DATA]VEJDATA.DAT',
     1    readonly,status='old')
c
      open(11,file='MSLUFT$MET:[HAVDATA.DATA]ROADEMI.DAT',
     1    status='new',carriagecontrol='list')
c
      open(12,file='MSLUFT$MET:[HAVDATA.DATA]R_COOR_MISS.DAT',
     1    status='new',carriagecontrol='list')
c
      open(13,file='MSLUFT$MET:[HAVDATA.FORTRAN]ROAD_TRAFFIC.ERR',
     1    status='new',carriagecontrol='list')
c
      do 10 I = 435,900,1
         do 20 J = 6045,6405,1
            OUT(I,J) = 0.
   20    continue
   10 continue
c
      do 15 I = 1,14,1
```

```
            A_REST(I) = 0.
   15   continue
c
        K       = 0
        SUM     = 0.
        W_SUM   = 0.
        EMISUM  = 0.
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Find the sum of traffic work and the number of records in the file  *
* from the Road Data Laboratory (file 10).                             *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        do 505 I = 1,20000,1
           read(10,100,end=510)AMT,ROAD,EXTRA,FROM,TO,DIST,
     1       SPEED,TRAFY,TRAFJ,WORK,FACL,FACR,BEGY,BEGX,ENDY,ENDX
           W_SUM = W_SUM + WORK
           K     = K + 1
  505   continue
  510   continue
c
        rewind 10
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Program start.                                                       *
* In the file (10) from the Road Data Laboratory is K records.         *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        do 500 I = 1,K,1
c
        AMT    = 0
        DIST   = 0
        EXTRA  = 0
        FACL   = 0
        FACR   = 0
        FLAG   = 0
        FROM   = 0
        ROAD   = 0
        SPEED  = 0
        TO     = 0
        TRAFJ  = 0
        TRAFY  = 0
c
        BEGX   = 0.
        BEGY   = 0.
        EMI    = 0.
        ENDX   = 0.
        ENDY   = 0.
        WORK   = 0.
c
        read(10,100,end=999)AMT,ROAD,EXTRA,FROM,TO,DIST,
     1      SPEED,TRAFY,TRAFJ,WORK,FACL,FACR,BEGY,BEGX,ENDY,ENDX
  100   format(1x,i3,i3,i1,2i8,i6,i4,2i6,f8.3,2i2,4f7.0)
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Skip records where no value for WORK is given                      *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        if (WORK .eq. 0) goto 500
c
        WORK = WORK * 1E3
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Write records where coordinates are missing and DIST > 1000        *
*   to file 12, for further investigations                             *
*   If DIST < 1000 and BEG or END coordinates are given, the           *
*   emission can be assined to a km2                                   *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
        if ((BEGY .eq. 0.) .or. (BEGX .eq. 0.) .or.
     1      (ENDY .eq. 0.) .or. (ENDX .eq. 0.)) then
           if (DIST .le. 1000) then
              if ((BEGX .ne. 0. .and. BEGY .ne. 0.) .or.
     1             (ENDX .ne. 0. .and. ENDY .ne. 0.)) then
```

```
                if (BEGX .ne. 0) then
                   goto 400
                else
                   BEGX = ENDX
                   ENDX = 0.
                   BEGY = ENDY
                   ENDY = 0.
                 endif
             else if (AMT .ne. 0) then
                  if (AMT .eq. 150) A_REST(1)  = A_REST(1)  + WORK
                  if (AMT .eq. 200) A_REST(2)  = A_REST(2)  + WORK
                  if (AMT .eq. 250) A_REST(3)  = A_REST(3)  + WORK
                  if (AMT .eq. 300) A_REST(4)  = A_REST(4)  + WORK
                  if (AMT .eq. 350) A_REST(5)  = A_REST(5)  + WORK
                  if (AMT .eq. 400) A_REST(6)  = A_REST(6)  + WORK
                  if (AMT .eq. 420) A_REST(7)  = A_REST(7)  + WORK
                  if (AMT .eq. 500) A_REST(8)  = A_REST(8)  + WORK
                  if (AMT .eq. 550) A_REST(9)  = A_REST(9)  + WORK
                  if (AMT .eq. 600) A_REST(10) = A_REST(10) + WORK
                  if (AMT .eq. 650) A_REST(11) = A_REST(11) + WORK
                  if (AMT .eq. 700) A_REST(12) = A_REST(12) + WORK
                  if (AMT .eq. 760) A_REST(13) = A_REST(13) + WORK
                  if (AMT .eq. 800) A_REST(14) = A_REST(14) + WORK
             else
                write(12,120)AMT,ROAD,EXTRA,FROM,TO,DIST,SPEED,
     1                 TRAFY,TRAFJ,WORK,FACL,FACR,BEGY,BEGX,ENDY,ENDX
             endif
          else if (AMT .ne. 0) then
                  if (AMT .eq. 150) A_REST(1)  = A_REST(1)  + WORK
                  if (AMT .eq. 200) A_REST(2)  = A_REST(2)  + WORK
                  if (AMT .eq. 250) A_REST(3)  = A_REST(3)  + WORK
                  if (AMT .eq. 300) A_REST(4)  = A_REST(4)  + WORK
                  if (AMT .eq. 350) A_REST(5)  = A_REST(5)  + WORK
                  if (AMT .eq. 400) A_REST(6)  = A_REST(6)  + WORK
                  if (AMT .eq. 420) A_REST(7)  = A_REST(7)  + WORK
                  if (AMT .eq. 500) A_REST(8)  = A_REST(8)  + WORK
                  if (AMT .eq. 550) A_REST(9)  = A_REST(9)  + WORK
                  if (AMT .eq. 600) A_REST(10) = A_REST(10) + WORK
                  if (AMT .eq. 650) A_REST(11) = A_REST(11) + WORK
                  if (AMT .eq. 700) A_REST(12) = A_REST(12) + WORK
                  if (AMT .eq. 760) A_REST(13) = A_REST(13) + WORK
                  if (AMT .eq. 800) A_REST(14) = A_REST(14) + WORK
          else
                write(12,120)AMT,ROAD,EXTRA,FROM,TO,DIST,SPEED,
     1                 TRAFY,TRAFJ,WORK,FACL,FACR,BEGY,BEGX,ENDY,ENDX
          endif
         goto 500
       endif
  120  format(1x,i3,i3,i1,2i8,i6,i4,2i6,f15.3,2i2,4f7.0)
c
  400  continue
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Find the FLAG, to determine which of the procedures in            *
*   subroutine SYS34_UTM is to be used                                *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       call FLAG_ROAD(AMT,ROAD,FLAG)
       if (FLAG .eq. 0) then
          call FAIL(13,'FLAG = 0 in FLAG_ROAD',AMT,ROAD)
       endif
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Transform the coordinates from system 34 to UTM Zone 32           *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       if (BEGX .ne. 0. .and. BEGY .ne. 0.)
     1 call SYS34_UTM(FLAG,BEGX,BEGY)
       if (ENDX .ne. 0. .and. ENDY .ne. 0.)
     1 call SYS34_UTM(FLAG,ENDX,ENDY)
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Calculate the NOx emission on actual roadpiece, from the          *
```

```
*    variable WORK and distribution of traffic and emission factors    *
*    given by Corinair                                                 *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         call NOX_EMI_ROAD(WORK,EMI)
         if (EMI .eq. 0)
      1  call FAIL(13,'EMI = 0 in NOX_EMI_ROAD',WORK,EMI)
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*    'Grid' the roadpiece and the calculated emission, on a UTM        *
*    Zone 32 grid, with reference to the southwestern corner of        *
*    the grid.                                                         *
*    In this subroutine values are also written to external file       *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        call GRID(BEGX,BEGY,ENDX,ENDY,EMI)
c
  500   continue
c
  999   continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Calculate the emission from the roads where no coordinates were      *
* and distribute the emission over the county                          *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
        do 25 I = 1,14,1
           if (A_REST(I) .ne. 0.) then
             if (I .eq. 1)  NR = '015'
             if (I .eq. 2)  NR = '020'
             if (I .eq. 3)  NR = '025'
             if (I .eq. 4)  NR = '030'
             if (I .eq. 5)  NR = '035'
             if (I .eq. 6)  NR = '040'
             if (I .eq. 7)  NR = '042'
             if (I .eq. 8)  NR = '050'
             if (I .eq. 9)  NR = '055'
             if (I .eq. 10) NR = '060'
             if (I .eq. 11) NR = '065'
             if (I .eq. 12) NR = '070'
             if (I .eq. 13) NR = '076'
             if (I .eq. 14) NR = '080'

          call REST(NR,A_REST(I))
        endif
   25   continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* After calculating the NOx emission from traffic on state- and        *
* county roads. Now find the NOx emission from municipal roads.        *
* The emission is distributed according to the inhabitants in          *
* each municipality and in the municipality after the inhabi-          *
* tants in town- and rual areas                                        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        call MUNIC_ROAD(W_SUM)
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Write the calculated emissions in each km2, from array OUT to        *
* external file. Output in tonne NOx a-1                               *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        do  30 I = 435,900,1
           do 40 J = 6045,6405,1
              if (OUT(I,J) .ne. 0) then
              SUM = SUM + OUT(I,J)/1000
              write(11,110) I,J,OUT(I,J)/1000
  110         format(i4,i5,f15.4)
             endif
   40      continue
   30   continue
c
        write(13,130) ('The sum of NOx emissions from road
      1 traffic is:'), SUM
  130   format(a,f15.4)
```

```
c
      write(13,*) ('The sum in EMISUM is: '), EMISUM/1000
c
      close(10)
      close(11)
      close(12)
      close(13)
c
      end
c
c
***********************************************************************
*                                                                     *
*   Fortran subroutine FLAG_ROAD                                      *
*   DMU-FOLU, 910117, Erik Runge.                                     *
*                                                                     *
*   Determine from the variables AMT and ROAD in the data from        *
*   the Road Data Laboratory which transformation procedure is to be  *
*   used in subroutine SYS34_UTM                                      *
*                                                                     *
*   Call: FLAG_ROAD(AMT,ROAD,FLAG)                                    *
*                                                                     *
*   Input:                                                            *
*     AMT   - The county, read from file                              *
*     ROAD  - The roads number from the Road Data Laboratory          *
*   Output:                                                           *
*     FLAG  - Indicating which procedure is to be used when           *
*             transforming coordinates from system 34                 *
*                                                                     *
***********************************************************************
c
      subroutine FLAG_ROAD(AMT,ROAD,FLAG)
c
      integer AMT,ROAD,FLAG
c
         if (AMT .eq. 0) then
            if (ROAD .le.  30) FLAG = 2
            if (ROAD .gt.  30 .and. ROAD .le.  79) FLAG = 1
            if (ROAD .gt.  79 .and. ROAD .le. 156) FLAG = 2
            if (ROAD .gt. 156 .and. ROAD .le. 176) FLAG = 3
            if (ROAD .gt. 176 .and. ROAD .le. 481) FLAG = 1
         endif
c
         if (AMT .ne. 0) then
            if (AMT .eq. 150) FLAG = 2
            if (AMT .eq. 200) FLAG = 2
            if (AMT .eq. 250) FLAG = 2
            if (AMT .eq. 300) FLAG = 2
            if (AMT .eq. 350) FLAG = 2
            if (AMT .eq. 400) FLAG = 3
            if (AMT .eq. 420) FLAG = 1
            if (AMT .eq. 500) FLAG = 1
            if (AMT .eq. 550) FLAG = 1
            if (AMT .eq. 600) FLAG = 1
            if (AMT .eq. 650) FLAG = 1
            if (AMT .eq. 700) FLAG = 1
            if (AMT .eq. 760) FLAG = 1
            if (AMT .eq. 800) FLAG = 1
         endif
       return
      end
c
c
**********************************************************************
*                                                                    *
* Fortran subroutine SYS34_UTM                                       *
*                                                                    *
* Transformation between coordiates in System 34 and                 *
* UTM coordinates.                                                   *
* made after the guidelines in:                                      *
* 'Saertryk af Landinspektoeren' VOL 30 Number 9 Auguse 1981         *
```

```
* title:  'Koordianttransformationer ved Geodaetisk Institut'          *
* Authers: O. Andersen and K. Poder, Geodaetisk Institut               *
*                                                                      *
* Call: subroutine SYS34_UTM(FLAG,X,Y)                                 *
*                                                                      *
* Input: FLAG - Determins which of the transformationroutines          *
*               is to be used.                                         *
*        1 - Transformation from system 34 to UTM,                     *
*            Jylland, Fyn and Langeland.                               *
*        2 - Transformation from system 34 to UTM,                     *
*            Sjælland, Lolland, Falster and Møn.                       *
*        3 - Transformation from system 34 to UTM,                     *
*            Bornholm                                                  *
*        4 - Transformation from UTM to system 34.                     *
*            Jylland, Fyn and Langeland.                               *
*        5 - Transformation from UTM to system 34.                     *
*            Sjælland, Lolland, Falster and Møn.                       *
*        6 - Transformation from UTM to system 34.                     *
*            Bornholm                                                  *
*          X    - double precision,                                    *
*                 the X coordinate to be transformed                   *
*          Y    - double precision                                     *
*                 the Y coordinate to be transformed                   *
* Output: X,Y the transformed coordinates, in meters                   *
*                                                                      *
* 16/10 - 1990 DMU, FOLU, Erik Runge                                   *
* Changed 910116, Erik Runge, To use the variable FLAG                 *
*                                                                      *
************************************************************************
c
      subroutine SYS34_UTM(FLAG,X,Y)
c
      double precision R(8),I(8),P,Q,Z,X,Y
      integer FLAG
c
c
     if (FLAG .eq. 1) then
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                         *
*  from system 34 to UTM Zone 32, Jylland, Fyn and Langeland           *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = -(X - 200000.)
            YM =    Y - 200000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1) =  6.2329350512D+06
            R(2) =  9.9950870832D-01
            R(3) = -2.2732469700D-11
            R(4) =  2.6264691305D-17
            R(5) =  0.0D+00
            R(6) =  0.0D+00
            R(7) =  0.0D+00
            R(8) =  0.0D+00
c
            I(1) =  5.9533558288D+05
            I(2) = -1.9984265541D-02
            I(3) = -3.3921905837D-10
            I(4) =  9.4810550416D-17
            I(5) =  0.0D+00
            I(6) =  0.0D+00
            I(7) =  0.0D+00
            I(8) =  0.0D+00
c
            do 400 T = 4,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
```

```
400         continue
c
            X = Q
            Y = P
            return
         endif
c
c
      if (FLAG .eq. 2) then
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                         *
*  from system 34 to UTM Zone 32.                                      *
*  Sjælland, Lolland, Falster and Møn.                                 *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = -(X - 200000.)
            YM =   Y - 200000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1)  =  6.2329256122D+06
            R(2)  =  9.9945508220D-01
            R(3)  = -7.3897597436D-11
            R(4)  = -5.2080308192D-17
            R(5)  =  0.0D+00
            R(6)  =  0.0D+00
            R(7)  =  0.0D+00
            R(8)  =  0.0D+00
c
            I(1)  =  5.9533534820D+05
            I(2)  = -1.9979251874D-02
            I(3)  = -2.0717946709D-09
            I(4)  =  2.5703781294D-16
            I(5)  =  0.0D+00
            I(6)  =  0.0D+00
            I(7)  =  0.0D+00
            I(8)  =  0.0D+00
c
            do 500 T = 4,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
500         continue
c
            X = Q
            Y = P
            return
       endif
c
c
      if (FLAG .eq. 3) then
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                         *
*  from system 45 to UTM Zone 33, Bornholm                             *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = -(X -  50000.)
            YM =   Y -  50000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1)  =  6.1074273382D+06
            R(2)  =  9.9959968336D-01
            R(3)  =  1.2072957024D-10
            R(4)  = -9.4593449880D-15
            R(5)  =  6.5994864072D-19
            R(6)  =  2.4410466467D-23
```

```
            R(7) = -2.8946629725D-27
            R(8) =  8.0282195772D-32
c
            I(1) =  4.9300725696D+05
            I(2) =  1.4686762273D-03
            I(3) =  5.4905973626D-10
            I(4) =  2.1510833240D-15
            I(5) = -1.0163477904D-18
            I(6) =  4.4726583308D-23
            I(7) = -1.1066966998D-27
            I(8) = -5.8658108998D-34
c
            do 600 T = 8,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
600         continue
c
            X = Q
            Y = P
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*   Transform from UTM Zone 33 to UTM Zone 32                        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
            call UTMUTM(X,Y,'3332',X_RET,Y_RET)
            X = X_RET
            Y = Y_RET
c
            return
          endif
c
c
      if (FLAG .eq. 4) then
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                       *
*  from UTM Zone 32 to system 34, Jylland, Fyn, Langeland            *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = X -  530000.
            YM = Y - 6231000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1) =  1.9937114647D+05
            R(2) =  1.0001365021D+00
            R(3) = -1.6486053260D-11
            R(4) = -1.8877421278D-17
            R(5) =  0.0D+00
            R(6) =  0.0D+00
            R(7) =  0.0D+00
            R(8) =  0.0D+00
c
            I(1) = -2.6538172775D+05
            I(2) =  1.9995520454D-02
            I(3) =  3.4452862838D-10
            I(4) = -9.6740792784D-17
            I(5) =  0.0D+00
            I(6) =  0.0D+00
            I(7) =  0.0D+00
            I(8) =  0.0D+00
c
            do 100 T = 4,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
100         continue
c
            X = -Q
```

```
            Y = P
            return
         endif
c
c
      if (FLAG .eq. 5) then
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                        *
*  from UTM Zone 32 to system 34,                                     *
*  Sjælland, Lolland, Falster and Møn.                                *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = X -  678000.
            YM = Y - 6131000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1) =  9.6432190004D+04
            R(2) =  9.9980074884D-01
            R(3) = -7.2077820180D-12
            R(4) =  6.2510854612D-17
            R(5) =  0.0D+00
            R(6) =  0.0D+00
            R(7) =  0.0D+00
            R(8) =  0.0D+00
c
            I(1) = -1.1935322229D+05
            I(2) =  1.9555849611D-02
            I(3) =  2.1675209626D-09
            I(4) = -2.5215002720D-16
            I(5) =  0.0D+00
            I(6) =  0.0D+00
            I(7) =  0.0D+00
            I(8) =  0.0D+00
c
            do 200 T = 4,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
200         continue
c
            X = -Q
            Y = P
            return
         endif
c
c
      if (FLAG .eq. 6) then
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*  Initializion of constants when transforming                        *
*  from UTM Zone 33 to system 45, Bornholm                            *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
c  Centroid displacement
            XM = X -  500000.
            YM = Y - 6100000.
c
            P = 0.0D+00
            Q = 0.0D+00
            R(1) =  4.2579934018D+04
            R(2) =  1.0004053319D+00
            R(3) = -7.7039294168D-11
            R(4) =  4.7460391692D-15
            R(5) =  1.8259975589D-18
            R(6) = -2.0727959960D-22
            R(7) =  7.0257273976D-27
            R(8) = -8.0529957628D-32
c
```

```
            I(1) = -4.2993547686D+04
            I(2) = -1.4643915438D-03
            I(3) = -4.7898338026D-10
            I(4) =  3.7432077762D-14
            I(5) = -4.8899686702D-18
            I(6) =  2.0540300545D-22
            I(7) = -2.9421011013D-27
            I(8) =  1.5387577766D-33
c
            do 300 T = 7,1,-1
            Z = P * YM - Q * XM + R(T)
            Q = P * XM + Q * YM + I(T)
            P = Z
300         continue
c
            X = -Q
            Y = P
            return
      endif
      end
c
c
      SUBROUTINE UTMUTM(X1R,X2R,DIR,Y1R,Y2R)
c
**********************************************************************
*                                                                    *
* Fortran subroutine UTMUTM                                          *
*                                                                    *
* Transformations between UTM Zone 32 and UTM Zone 33               *
* made after the guidelines in:                                      *
* 'Saertryk af Landinspektoeren' VOL 30 Number 9 Auguse 1981         *
* title:  'Koordianttransformationer ved Geodaetisk Institut'        *
* Authers: O. Andersen and K. Poder, Geodaetisk Institut             *
*                                                                    *
* Call:  subroutine UTMUTM(X1R,X2R,DIR,Y1R,Y2R)                      *
*                                                                    *
* Input: X1R - the X coordinate to be transformed                    *
*        X2R - the Y coordinate to be transformed                    *
*        DIR - a logical, giving the 'direction' of the              *
*              transformation;                                       *
*              DIR = '3233' - transformation from Zone 32 to 33      *
*              DIR = '3332' - transformation from Zone 33 to 32      *
*                                                                    *
* Output: Y1R - the transformed X coordinate                         *
*         Y2R - the transformed Y coordinate                         *
*                                                                    *
* 12/10 - 1990 DMU, FOLU, Lisbeth Stenfalk & Erik Runge              *
*                                                                    *
**********************************************************************
c
      double precision X1R,X2R,Y1R,Y2R
c
      real*16 A,B,DI(9),DIFX1,DIFX2,P,Q,R(9),TOL,
     1       X1,X1B,X1M,X2,X2B,X2M,
     2       Y1,Y1M,Y2,Y2M,Z
c
      character*4 DIR
c
c Initialisations
c
      R(1)  =   16257.546929D00
      R(2)  =      -2.065074625D-3
      R(3)  =      -7.970160627D-10
      R(4)  =       3.516381368D-17
      R(5)  =       6.373566966D-24
      R(6)  =      -1.960680878D-31
      R(7)  =      -1.840598563D-38
      R(8)  =       7.101938265D-46
      R(9)  =       3.000000000D-53
c
      DI(1) = -373968.399798D00
```

```
      DI(2) =        8.695605083D-2
      DI(3) =        4.528930217D-9
      DI(4) =       -3.604212207D-16
      DI(5) =       -7.724604689D-24
      DI(6) =        4.693808792D-31
      DI(7) =       -5.246214889D-39
      DI(8) =       -3.130102892D-46
      DI(9) =        5.000000000D-53
c
      X1 = X1R
      X2 = X2R
c
      if (DIR.eq.'3233') go to 20
c
      if (DIR.ne.'3332') go to 200
c
      do 10 I = 1,9
        DI(I) = -1.0*DI(I)
   10 continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c Program start                                                        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   20 TOL    = 0.0004
c
      P      = 0.0
      Q      = 0.0
c
      X1M    = X1 -  500000.000000
      X2M    = X2 - 6206216.606720
c
      do 50 J = 1,9
       I     = 10 - J
       Z     = P*X2M - Q*X1M + R(I)
       Q     = P*X1M + Q*X2M + DI(I)
       P     = Z
   50 continue
c
      Y2     = X2 + P
      Y1     = X1 + Q
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Backtransformation for control of calculations                       *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
      P      = 0.0
      Q      = 0.0
c
      Y1M    = Y1 -  500000.000000
      Y2M    = Y2 - 6206216.606720
c
      do 60 J = 1,9
       I     = 10 - J
       Z     = P*Y2M - Q*Y1M + R(I)
       Q     = P*Y1M + Q*Y2M - DI(I)
       P     = Z
   60 continue
c
      X2B    = Y2 + P
      X1B    = Y1 + Q
c
      DIFX1 = QABS(X1B - X1)
      DIFX2 = QABS(X2B - X2)
c
      if((DIFX1.gt.TOL.or.DIFX2.gt.TOL).and.DIR.eq.'3233')
     1 write(13,100) X1,X2,DIFX1,DIFX2
  100 format(/2X,'Zone 32 coordinates (',F15.5,',',F15.5,') does not',
     1      ' backtransform within the tolerance 0.0004',/,
     2      '   DIFX1 = ',F15.5,'    DIFX2 = ',F15.5)
c
      if((DIFX1.gt.TOL.or.DIFX2.gt.TOL).and.DIR.eq.'3332')
```

```
    1 write(13,101) X1,X2,DIFX1,DIFX2
  101 format(/2X,'Zone 33 coordinates (',F15.5,',',F15.5,') does not',
     1     ' backtransform within the tolerance 0.0004',/,
     2     '   DIFX1 = ',F15.5,'    DIFX2 = ',F15.5)
c
      Y1R = Y1
      Y2R = Y2
c
      go to 300
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c Message in case of error in the parameter 'DIR'                      *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c
  200 write(13,220) DIR
  220 format(2X,'You have called a wrong subroutine or your',/,
     1        2X,'direction parameter DIR = ',A4,' is wrong.')
c
  300 return
      end
c
c
***********************************************************************
*                                                                     *
*   Fortran subroutine NOX_EMI_ROAD                                   *
*   DMU-FOLU, 910117, Erik Runge                                      *
*                                                                     *
*   Calculates the NOx emission on a road piece. From the             *
*   traffic work (WORK) on that road piece (given by Road             *
*   Data Laboratory) and distribution of traffic and                  *
*   emission factores given by Corinair.                              *
*                                                                     *
*   Call: NOX_EMI_ROAD(WORK,EMI)                                      *
*                                                                     *
*   Input:                                                            *
*     WORK    - The average mill. car kilometers per year on          *
*               the road piece.                                       *
*   Output:                                                           *
*     EMI     - The NOx emission on the road                          *
*                                                                     *
*   Auxilary ariables:                                                *
*     K       -                                                       *
*     FRACTO  - Holds the sum of the values in array FRAC             *
*                                                                     *
*   Arrays:                                                           *
*     FRAC    - The distribution of different traffic types           *
*               in mia. waggon km per. year (1986).                   *
*               Information from Corinair                             *
*         1   - Passengercars and light vans, petrol                  *
*         2   - Heavy vans, petrol                                    *
*         3   - Diesel vehicle under 3500 kg                          *
*         4   - Diesel vehicle over  3500 kg                          *
*     EMIFAC  - NOx emission factors in g per. km                     *
*               Information from Corinair                             *
*               Type of vehicle as mentioned under array FRAC         *
*                                                                     *
***********************************************************************
c
      subroutine NOX_EMI_ROAD(WORK,EMI)
c
      integer K
      double precision FRAC,FRACTO,EMI,EMIFAC,J,WORK,
     1                 A_REST,OUT,EMISUM
      dimension FRAC(1:4),EMIFAC(1:4)
      common A_REST(1:14),OUT(435:900,6045:6405),EMISUM
c
      K      = 0
      FRACTO = 0.
      J      = 0.
c
         do 500 K = 1,4,1
```

```
            FRACTO = FRACTO + FRAC(K)
  500     continue
c
          J      = WORK/FRACTO
          EMI    = (EMIFAC(1)*FRAC(1)*J + EMIFAC(2)*FRAC(2)*J +
     1               EMIFAC(3)*FRAC(3)*J + EMIFAC(4)*FRAC(4)*J)
c
      EMISUM = EMISUM + EMI
c
          data FRAC/22.14,0.44,6.55,2.78/
          data EMIFAC/2.1,6.0,0.67,17.39/
         return
      end
c
c
***********************************************************************
* Fortran subroutine GRID                                             *
* DMU-FOLU, 910207, Erik Runge                                        *
*                                                                     *
* Grids calculated emissions on a UTM grid.                           *
*                                                                     *
* Call: GRID(X_OLD,Y_OLD,X,Y,EMI)                                     *
* Input: X_OLD,Y_OLD - The UTM X and Y coordinate of the starting     *
*                      point in km.                                   *
*        X,Y         - The UTM X and Y coordinate of the ending       *
*                      point in km.                                   *
*        EMI         - The emission on the distance from X_OLD,Y_OLD  *
*                      to X,Y                                         *
* Output: The gridded emissions are written in common array OUT       *
*                                                                     *
***********************************************************************
      subroutine GRID(X_OLD,Y_OLD,X,Y,EMI)
c
      integer MINX,MINY
      real*8  BEGX,ENDX,STEPX,X,X_OLD,
     1        BEGY,ENDY,STEPY,Y,Y_OLD,
     1        A_REST,EMI,EMISUM,PEMI,OUT,T,J1
      common A_REST(1:14),OUT(435:900,6045:6405),EMISUM
c
      BEGX = X_OLD/1000.
      ENDX = X/1000.
      BEGY = Y_OLD/1000.
      ENDY = Y/1000.
      J1   = 0.
      j2   = j2 + 1
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Determine the southwestern corner of the grid element, as a        *
* reference to that grid element.                                    *
* Note that MINX and MINY are integers.                              *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      MINX = BEGX
      MINY = BEGY
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* The subroutine might be called with ENDX = 0 and ENDY = 0, but only *
* if the distance is less than 1000 m. In theese cases the emission is*
* connected to the grid element of BEGX,BEGY                         *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      if (ENDX .eq. 0. .and. ENDY .eq. 0.) then
          OUT(MINX,MINY) = OUT(MINX,MINY) + EMI
          return
      endif
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the actual distance is not inside one gridelement, then step    *
* until the reference point is not longer the same.                  *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      if (int(ENDX) .ne. MINX .or. int(ENDY) .ne. MINY) then
      do 200 T = 1.,100000.,1.
          STEPX = BEGX + T/100000. * (ENDX - BEGX)
          STEPY = BEGY + T/100000. * (ENDY - BEGY)
```

```
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Calculate the part of the emission in this gridelement and write    *
* the value in array OUT.                                              *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
          if (int(STEPX) .ne. MINX .or. int(STEPY) .ne. MINY) then
              PEMI = EMI * (T-J1)/100000.
              OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
              MINX    = STEPX
              MINY    = STEPY
              J1 = T
          endif
  200  continue
       endif
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the distance or the rest of the distance after stepping, is      *
* inside one grid element, then calculate the emission and write      *
* the value in array OUT.                                              *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       if (int(ENDX) .eq. MINX .and. int(ENDY) .eq. MINY) then
           PEMI = EMI * (1 - J1/100000.)
            OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
       endif
       return
       end
c
c
**********************************************************************
* Fortran subroutine REST                                             *
* 910125, DMU-FOLU, Erik Runge                                        *
*                                                                     *
* In the data from the Road Data Laboratory some coordinates are mis- *
* sing in these cases and when the county code is given the calculated*
* NOx emission is in this subroutine distributed over the county.     *
* Call: REST(NR,WORK)                                                 *
*                                                                     *
* Input: NR   - The number of the county                              *
*        WORK - The traffic work/year in mia km, summerized from data *
*               given by the Road Data Laboratory.                    *
*                                                                     *
* The results are added to the common array OUT                       *
*                                                                     *
* The variables:                                                      *
*    PART  - The part of the actual km2 covered by the county         *
*    SUM   - The sum of PART's within one county                      *
*    X,Y   - UTM Zone 32 coordinates to the southwestern corner       *
*            of the grid element                                      *
*    EMI   - The total calculated NOx emission in actual county       *
*    PEMI  - The emission per PART                                    *
*    XYEMI - The calculated NOx emission in actual km2                *
*                                                                     *
* Subroutines called:                                                 *
*    Subroutine NOX_EMI_ROAD                                          *
*                                                                     *
**********************************************************************
c
       subroutine REST(NR,WORK)
c
       character NR*3, NFILE*52
       integer X,Y
       double precision A_REST,EMI,OUT,PART,PEMI,SUM,WORK,XYEMI,
     1                  EMISUM
       real BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
c
       common A_REST(1:14),OUT(435:900,6045:6405),EMISUM
c
       call NOX_EMI_ROAD(WORK,EMI)
       if (EMI .eq. 0.)
     1 call FAIL(13,'EMI = 0 in NOX_EMI_ROAD',WORK,EMI)
c
```

```
      NFILE = 'msluft$met:[havdata.kommune]AMT'//NR//'.DAT'
c
      open(10,file=NFILE,status='old')
c
 600  read(10,200,end=700)
     1       X,Y,BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
      SUM = SUM + BO + FP + S + E + M + MO + MA + H + SA + FV
      goto 600
 700  rewind 10
c
      PEMI = EMI/SUM
c
 800  read(10,200,end=999)
     1       X,Y,BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
      SUM = BO + FP + S + E + M + MO + MA + H + SA + FV
c
      XYEMI = PEMI * SUM
c
      if (XYEMI .ne. 0.) then
         if ((X .lt.  435 .or. X .gt.  900) .or.
     1        (Y .lt. 6045 .or. Y .gt. 6405)) then
         call FAIL(13,' X or Y out of range (X,Y)(REST):',MINX,MINY)
         else
            OUT(X,Y) = OUT(X,Y) + XYEMI
         endif
      endif
c
      goto 800
c
 200  format(1x,i3,i5,13f5.2)
c
 999  continue
c
      close(10)
c
      return
c
      end
c
c
*******************************************************************
*                                                                 *
* Fortran subroutine MUNIC_ROAD (NOx emission)                    *
* DMU-FOLU, 901127, Erik Runge                                    *
*                                                                 *
* In this subroutine the NOx emission on the municipal roads is cal-  *
* culated from the traffic work - given by the Road Data Laboratory   *
* and emissionfactors given by Corinair.                          *
* The total traffic work is distributed over the country after the    *
* number of inhabitants in each municipality and on the municipality  *
* after inhabitants in town- and rualareas. Information from the      *
* Danish Statistical office,1988.                                 *
* OBS - to run this subroutine, there have to be files - for each     *
* municipality, made on basis of the 'land use' registration, with    *
* information of X and Y coordinates and the 13 'land use' categories *
*                                                                 *
* Call; MUNIC_ROAD(W_SUM)                                         *
*                                                                 *
* The variables;                                                  *
*    AID_R  - Holds the NOx emission per part of rual (field) area of *
*             the municipality                                    *
*    AID_T  - Holds the NOx emission per part of inhabited area of    *
*             the municipality                                    *
*    EMIS   - The caluclated NOx emission in actual km2           *
*    NR     - The number of the municipality, read from file 10   *
*    POPULA - The total population in Denmark 1st of January 1988 *
*             (From the Danish Staticticał Office)                *
*    RES_R  - The NOx emission in a municipality's rual area      *
*    RES_T  - The NOx emission in a municipality's town area      *
*    SUM_BO - The sum of inhabited area of the municipality, as   *
*             registreted in 'land_use'                           *
```

```
*     SUM_M  - The sum of rual (field) area of the municipality, as     *
*               registreted in 'land_use'                                *
*     TOTAL  - Total number of inhabitants in each municipality.        *
*               Read from file 20.                                       *
*     TOWN   _ Number of inhabitants in the town area of the munic.     *
*     WORK   - The total trafficwork (trafikarbejde) (31.91 mia km) on *
*               all Danish roads 1988 in waggonkilometers (vognkm)       *
*               (As used by Corinair for 1985)                           *
*               Minus the work summerized from file 10 in mainprogram   *
*               (W_SUM). WORK holds thereby the traficwork on municipal *
*               roads.                                                   *
*     WORK_R - The trafficwork in the rual area of the municipality     *
*     W_SUM  _ The sum of traffic work on state- and county roads,      *
*               given by the Road Data Laboratory                        *
*     WORK_T - The trafficwork in the town area of the municipality     *
*     WORKPO - The trafficwork per inhabitant                            *
*     X,Y    - X and Y coordinates read from file 30                    *
*     BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV - The categories as used in     *
*               the 'land use' registration                              *
*                                                                        *
* The files;                                                             *
*    File 20 - MSLUFT$MET:[HAVDATA.KOMMUNE]POP1.DAT                      *
*               Contains information on the population in each          *
*               municipality - total and in towns.                      *
*              (the Danish Statistical Office)                          *
*    File 30 - MSLUFT$MET:[HAVDATA.DATA]'KOMM_'//NR//'.DAT' (KFILE)     *
*               Files created on basis of the 'Land use in Denmark'     *
*               registration - one file for each municipality.          *
*               Information in UTM km2 on what the area is used to,     *
*               among other also the inhabited- and the field area.     *
*               Used to distribute the NOx emission on UTM km2          *
*                                                                        *
* Subroutines called;                                                    *
*    subroutine NOX_EMI_ROAD                                             *
*                                                                        *
**************************************************************************
c
      subroutine MUNIC_ROAD(W_SUM)
c
      character KFILE*52, NR*3
      integer RUAL,TOTAL,TOWN,X,Y
      double precision A_REST,OUT,POPULA,RES_T,RES_R,WORK,WORKPO,
     1      W_SUM,WORK_R,WORK_T,AID_R,AID_T,EMIS,SUM_BO,SUM_M,
     1      EMISUM
      real BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
c
      common A_REST(1:14),OUT(435:900,6045:6405),EMISUM
c
      POPULA  = 5129254
      WORK    = (31910 - W_SUM) * 1E3
      WORKPO  = WORK/POPULA
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* File 20 has information of number, total number of people and        *
* number of people in inhabited areas of each municipality             *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      open(20,file='MSLUFT$MET:[HAVDATA.KOMMUNE]POP1.DAT',
     1      status='old')
c
  500 continue
c
      close(30)
      AID_R  = 0.
      AID_T  = 0.
      EMIS   = 0.
      SUM_BO = 0.
      SUM_M  = 0.
      NR     = ' '
      KFILE  = ' '
c
      read(20,201,end=998) NR,TOTAL,TOWN
```

```
  201   format(a,2i7)
c
        RUAL = TOTAL - TOWN
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Find the trafficwork in inhabited (WORK_T) and in rual             *
* (WORK_R) areas                                                     *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        WORK_T = WORKPO * TOWN
        WORK_R = WORKPO * RUAL

*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Calcualate the emission from traffic work in inhabited (RES_T)     *
* and rual (RES_R) areas                                             *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        call NOX_EMI_ROAD(WORK_T,RES_T)
        call NOX_EMI_ROAD(WORK_R,RES_R)
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Open the file (KFILE) with data about the acutal municipality (NR).*
* Data from the 'land use' registration.                             *
* Summerize the values of inhabited (BO) and rual (field) (M) areas. *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        KFILE = 'msluft$met:[havdata.kommune]komm_'//NR//'.dat'
        open(30,file=KFILE,status='old')
c
  600   read(30,300,end=700) X,Y,BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
           SUM_BO = SUM_BO + BO
           SUM_M  = SUM_M  + M
        goto 600
c
  700   continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* The NOx emission per inhabited 1/4 km2 of the municipality         *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        if (SUM_BO .ne. 0) then
          AID_T = RES_T/SUM_BO
        else
          AID_T = 0.
        endif
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* The NOx emission per field (rual) 1/4 km2 of the municipality      *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        if (SUM_M .ne. 0) then
          AID_R = RES_R/SUM_M
        else
          AIDR = 0.
        endif
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Rewind the file with informations about the actual municipality    *
* and distribute the calculated NOx emission over the km2 the munici- *
* pality covers according to the registrated urban- and rual areas   *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        rewind 30
c
  601   read(30,300,end=500) X,Y,BO,FP,S,L,N,E,M,MO,MA,H,S,FV,HV
c
        EMIS = (BO * AID_T + M * AID_R)
c
           if ((X .lt.  435 .or. X .gt.  900) .or.
     1          (Y .lt. 6045 .or. Y .gt. 6405)) then
                call FAIL(13,' X or Y out of range (MUNIC_ROAD):',X,Y)
           else
                OUT(X,Y) = OUT(X,Y) + EMIS
           endif
c
        goto 601
c
```

```
  300   format(1x,i3,i5,13f5.2)
c
  998   continue
c
        close(20)
        close(30)
c
        return
        end
c
c
***********************************************************************
*                                                                     *
*   Fortran subroutine FAIL                                           *
*   DMU-FOLU, 910117, Erik Runge.                                     *
*                                                                     *
*   Writes a message to a file if an call of a subroutine             *
*   returnes with an unexpected answer                                *
*                                                                     *
*   Call: FAIL(F_FILE,TEXT,A,B)                                       *
*                                                                     *
*   Input:                                                            *
*     F_FILE - The file to which the message is written               *
*     TEXT   - The text written to file F_FILE                        *
*     P1,P2  - Paramters,                                             *
*              used to indicate where the error occured               *
*                                                                     *
***********************************************************************
c
        subroutine FAIL(F_FILE,TEXT,P1,P2)
c
        integer F_FILE
        real P1,P2
        character*(*) TEXT
c
           write(F_FILE,130)TEXT,P1,P2
  130      format(a,2f15.4)
c
         return
        end
```

```
***********************************************************************
*                                                                     *
* Fortran program NOX_CORINE                                          *
* DMU-FOLU, 901123, Erik Runge                                        *
*                                                                     *
* Distributes the on each county (amt), in Corinair, EMI recorded     *
* NOx emissions, on the 1x1 km2 of the county. With reference to      *
* the southwestern corner of each UTM Zone 32 grid element.           *
*                                                                     *
* To run the program it is necessary to have, for each municipality   *
* data based on the 'Land use' registration.                          *
*                                                                     *
* Variables:                                                          *
*   NR - The number of the municipality. A char in order to be able   *
*        to concatinate the value for the open file statement.        *
*   NFILE  - The file to be opened.                                   *
*   AMT    - The county ('amt' in Danish).                            *
*   K_POP  - The number of inhabitants in the municipality.           *
*   K_RUAL - Number of inhabitants not in towns in the municipality.  *
*   K_TOWN - Number of inhabitants in towns in the municipality.      *
*   X,Y,BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV -                            *
*      Categories from the 'land use' registration.                   *
*   SUM_BO - Sum of variable BO for each municipality.                *
*   SUM_M  - Sum fo variable M  for each municipality.                *
*                                                                     *
* Arrays:                                                             *
*    A_POP  - holds the number of inhabitants of each county          *
*              1 - København and Frederiksberg                        *
*              2 - Københavns amt (015)                               *
*              3 - Frederiksborg amt (020)                            *
*              4 - Roskilde amt (025)                                 *
*              5 - Vestsjællands amt (030)                            *
*              6 - Storstrøms amt (035)                               *
*              7 - Bornholms amt (040)                                *
*              8 - Fyns amt (042)                                     *
*              9 - Sønderjyllands amt (050)                           *
*             10 - Ribe amt (055)                                     *
*             11 - Vejle amt (060)                                    *
*             12 - Ringkøbing amt (065)                               *
*             13 - Århus amt (070)                                    *
*             14 - Viborg amt (076)                                   *
*             15 _ Nordjyllands amt (080)                             *
*                                                                     *
*    E_COUN - holds the emission from Corinair in the couties         *
*              1. dimension is a reference to the county,             *
*                 as explained under A_POP                            *
*              2. dimension;                                          *
*                 NOx emission in Mg/year                             *
*                   1 - from combustion (domestic heating)            *
*                   2 - from energy generation                        *
*                   3 - from combustion in industry                   *
*                   4 - from industrial production                    *
*                   5 - from traffic                                  *
*                                                                     *
*    E_POP  - holds the emission in the 15 counties and               *
*              5 categories per inhabitant in that county             *
*                                                                     *
*    OUT    - Holds the calculated gridded emission values            *
*              1. dimension is the X coordinate                       *
*              2. dimension is the Y coordinate                       *
*              3. dimension;                                          *
*                 NOX emission in Mg/year                             *
*                   1 - from combustion (domestic heating)            *
*                   2 - from energy generation                        *
*                   3 - from combustion in industry                   *
*                   4 - from industrial production                    *
*                   5 - from traffic                                  *
*                   6 - sum of 1,2,3,4                                *
*                                                                     *
* Subroutines called:                                                 *
*    FIND_COUNTY(NR,AMT)                                              *
```

```
*     READY(UNIT,FILE,LINES)                                              *
*                                                                         *
***************************************************************************
c
       program NOX_CORINE
c
       character NR*3,NFILE*52
       integer AMT,K_POP,K_RUAL,K_TOWN,A_POP,E_COUN,X,Y
       real BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV,SUM_BO,SUM_M
       double precision E_POP,OUT
       dimension A_POP(1:15),E_COUN(1:15,1:5),E_POP(1:15,1:5),
     1          OUT(435:900,6045:6405,1:6)
c
       data A_POP/553177,602046,340513,215993,283271,
     1            257007, 46105,458111,250158,218460,
     1            329847,266834,594184,230318,483754/
c
       data E_COUN/ 912,  985, 493, 307, 495, 488,  125, 526,
     1              386,  222, 475, 358, 634, 390,  640,
     1              361, 1174, 637, 415,  327, 346, 109, 508,
     1              551,  278,1024,2605, 2576, 709,1280,
     1             1344, 1453, 824, 572,  555, 492,  75,1141,
     1              655,  583,1109, 921, 1407, 566,1041,
     1              164,  177, 337,  61,   67, 157,  29,  86,
     1               67,   54, 150,  69,  154,  90,2367,
     1             6818,10236,6122,5110, 6473,4800, 911,9160,
     1             5930, 5001,7863,6212,12049,5432,10714/
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Initialisations                                                        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       do 900 J = 1,5,1
       do 800 I = 1,15,1
          if (A_POP(I) .ne. 0) then
             E_POP(I,J) = float(E_COUN(I,J))/float(A_POP(I))
          else
             E_POP(I,J) = 0.
          endif
  800  continue
  900  continue
c
       do 10 I = 435,900,1
          do 20 J = 6045,6405,1
             do 30 K = 1,6,1
                OUT(I,J,K) = 0.
   30        continue
   20     continue
   10  continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Open files.Subroutine READY opens file 11 and skip the first 12 lines+
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       call READY(11,'msluft$met:[havdata.kommune]POPULATION.DAT',12)
c
       open(14,file='msluft$met:[havdata.data]NOX_EMI.DAT',
     1      status='new',carriagecontrol='list')
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Do for each record in file 11, ie for each municipality               *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  500  continue
       close(12)
       AMT    = 0
       K_RUAL = 0
       NFILE  = ' '
       NR     = ' '
       SUM_BO = 0.
       SUM_M  = 0.
c
       read(11,110,end=999) NR, K_POP, K_TOWN
  110  format(a,2i7)
```

```
c
      K_RUAL = K_POP - K_TOWN
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Open the file for this municipality, based on the 'Land use' regist. *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      NFILE = 'msluft$met:[havdata.kommune]komm_'//NR//'.dat'
      call READY(12,NFILE,0)
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Summarize the build-up and rual (field) areas                        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  600 read(12,*,end=700) X,Y,BO,FP,S,L,N,E,M,MO,MA,H,SA,FV,HV
         SUM_BO = SUM_BO + BO
         SUM_M  = SUM_M  + M
      goto 600
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* For each pair of X,Y coordinates in this municipal file, calculate    *
* the emission. Category 1 (heating) and 5 (traffic) are distributed    *
* after the number of inhabitants in towns and not in towns.            *
* Category 2, 3 and 4 are assumed to be placed in towns and are         *
* only distributed over build-up areas.                                 *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  700 continue
      rewind 12
c
      call FIND_COUNTY(NR,AMT)
c
  601 read(12,*,end=500) X,Y,BO,FP,S,L,N,E,M,MO,MA,H,S,FV,HV
c
      if (SUM_BO .ne. 0) then
       OUT(X,Y,1) = OUT(X,Y,1) + BO * (E_POP(AMT,1) * K_TOWN / SUM_BO)
       OUT(X,Y,2) = OUT(X,Y,2) + BO * (E_POP(AMT,2) * K_POP  / SUM_BO)
       OUT(X,Y,3) = OUT(X,Y,3) + BO * (E_POP(AMT,3) * K_POP  / SUM_BO)
       OUT(X,Y,4) = OUT(X,Y,4) + BO * (E_POP(AMT,4) * K_POP  / SUM_BO)
       OUT(X,Y,5) = OUT(X,Y,5) + BO * (E_POP(AMT,5) * K_TOWN / SUM_BO)
      endif
c
      if (SUM_M .ne. 0) then
       OUT(X,Y,1) = OUT(X,Y,1) + M  * (E_POP(AMT,1) * K_RUAL / SUM_M)
       OUT(X,Y,5) = OUT(X,Y,5) + M  * (E_POP(AMT,5) * K_RUAL / SUM_M)
      endif
c
      goto 601
c
  999 continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Summarize each record. But leave out emissions from traffic (OUT(5)) *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      do 50 I = 435,900,1
         do 60 J = 6045,6405,1
            OUT(I,J,6) = OUT(I,J,1)+OUT(I,J,2)+OUT(I,J,3)+OUT(I,J,4)
   60    continue
   50 continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Write the values to external file. Excluding traffic (OUT(5))        *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     do 51 J = 6045,6405,1
         do 61 I = 435,900,1
            if (OUT(I,J,6) .gt. 0.) then
               write(14,140)
     1   I,J,OUT(I,J,1),OUT(I,J,2),OUT(I,J,3),OUT(I,J,4),OUT(I,J,6)
            endif
   61    continue
   51 continue
  140 format(i3,i5,5f11.5)
c
      close(11)
```

```
      close(12)
      close(14)
c
      end
c
c
******************************************************************
*                                                                *
* Fortran subroutine FIND_COUNTY(NR,AMT)                         *
* 901123, DMU-FOLU, Erik Runge                                   *
*                                                                *
* From the number of the municipality (kommune), NR this         *
* subroutine returns the number of the county (amt), AMT         *
*                                                                *
* In the return, if AMT =                                        *
*             1 - København and Frederiksberg                    *
*             2 - Københavns amt (015)                           *
*             3 - Frederiksborg amt (020)                        *
*             4 - Roskilde amt (025)                             *
*             5 - Vestsjællands amt (030)                        *
*             6 - Storstrøms amt (035)                           *
*             7 - Bornholms amt (040)                            *
*             8 - Fyns amt (042)                                 *
*             9 - Sønderjyllands amt (050)                       *
*            10 - Ribe amt (055)                                 *
*            11 - Vejle amt (060)                                *
*            12 - Ringkøbing amt (065)                           *
*            13 - Århus amt (070)                                *
*            14 - Viborg amt (076)                               *
*            15 _ Nordjyllands amt (080)                         *
*                                                                *
******************************************************************
c
      subroutine FIND_COUNTY(NR,AMT)
c
      character NR*3
      integer AMT
c
       if (NR .eq. '101' .or.  NR .eq. '147') AMT  =  1
       if (NR .ge. '151' .and. NR .le. '189') AMT  =  2
       if (NR .ge. '201' .and. NR .le. '237') AMT  =  3
       if (NR .ge. '251' .and. NR .le. '271') AMT  =  4
       if (NR .ge. '301' .and. NR .le. '345') AMT  =  5
       if (NR .ge. '351' .and. NR .le. '397') AMT  =  6
       if (NR .ge. '401' .and. NR .le. '411') AMT  =  7
       if (NR .ge. '421' .and. NR .le. '499') AMT  =  8
       if (NR .ge. '501' .and. NR .le. '545') AMT  =  9
       if (NR .ge. '551' .and. NR .le. '577') AMT  = 10
       if (NR .ge. '601' .and. NR .le. '631') AMT  = 11
       if (NR .ge. '651' .and. NR .le. '685') AMT  = 12
       if (NR .ge. '701' .and. NR .le. '751') AMT  = 13
       if (NR .ge. '761' .and. NR .le. '793') AMT  = 14
       if (NR .ge. '801' .and. NR .le. '861') AMT  = 15
      return
      end
```

```
***********************************************************************
* Fortran program SHIPS                                               *
* 910205, DMU-FOLU, Erik Runge                                        *
*                                                                     *
* This program calculates the NOx emission on the two main shiproutes *
* through Danish waters. The routes are; T (transit) from Skagen,     *
* through Storebælt, south of Lolland and Falster and into Østersøen. *
* D - B from Skagen to east of Anholt the same as route T, but from   *
* Anholt this routes goes through the Sound into the Baltic Sea.      *
*                                                                     *
* The calculated emissions (in tonne NOx/year) are distributed on a   *
* 1x1 km2 UTM Zone 32 grid, for the use in dispersion models.         *
*                                                                     *
* The variables;                                                      *
*    DIST  - The distance in meters                                   *
*    EMI   - The calculated emission on actual route or part of route. *
*    NEW   - Values used in Uniras plotting program, 2 ~ 'hold        *
*            pen down' (one continous line), 3 ~ 'lift pen up and     *
*            put it down'.                                            *
*    NO    - A number of the route, used to determine which emis-     *
*            sion factor is to be used.                               *
*    X_OLD,Y_OLD,X,Y - The starting and ending points of part of the  *
*            shiproute.                                               *
* Arrays;                                                             *
*    OUT   - Holds the calculated NOx emission each UTM Zone 32       *
*            grid elements.                                           *
* Files;                                                              *
*    File 10 - Holds the coordinates, in UTM Zone 32, to the ship     *
*              routes.                                                *
*    File 13 - The 'error' file to which errors - if any - is         *
*              written during program execution.                      *
*    File 20 - Is the 'output' file, to which the calculated NOx      *
*              emissions are written.                                 *
*                                                                     *
* Subroutines called;                                                 *
*    Subroutine NOX_EMI                                               *
*    Subroutine GRID                                                  *
*    Subroutine FAIL                                                  *
*                                                                     *
***********************************************************************
c
      program SHIPS
c
      integer NEW,NO
      real*8 DIST,EMI,OUT,X,X_OLD,Y,Y_OLD
      common OUT(570:770,6030:6420)
c
      open(10,file='msluft$met:[havdata.kortplot]MAIN_SHIP_ROUTES.DAT',
     1        readonly,status='OLD')
c
      open(13,file='msluft$met:[havdata.fortran]SHIPS.ERR',
     1        status='NEW',carriagecontrol='LIST')
c
      open(20,file='msluft$met:[havdata.data]SHIPS_NOX_EMI.DAT',
     1        status='NEW',carriagecontrol='LIST')
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Read a record, if it is part of the same route (NEW = 2) then       *
* calculate and grid the emission. If the record read is start of     *
* a new route then save the X,Y coordinates and read the next record. *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  500 continue
      X_OLD = X
      Y_OLD = Y
      read(10,110,end=999)X,Y,NEW,NO
      if (NEW .eq. 2) then
         DIST = sqrt((X-X_OLD)**2 + (Y-Y_OLD)**2)
         call NOX_EMI(DIST,NO,EMI)
         if (EMI .le. 0.) call FAIL(13,'EMI is .le. 0 after NOX_EMI',
     1   DIST,EMI)
         call GRID(X_OLD,Y_OLD,X,Y,EMI)
```

```
         goto 500
      else
         goto 500
      endif
c
 110  format(1x,2f10.5,i3,26x,i2)
 999  continue
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Write the gridded calculated emissions to external file               *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      do  30 I = 570,770,1
         do 40 J = 6030,6420,1
            if (OUT(I,J) .ne. 0) then
            write(20,220) I,J,OUT(I,J)
 220        format(i4,i5,f15.4)
           endif
  40     continue
  30  continue
c
      end
c
c
**************************************************************************
* Fortran subroutine NOX_EMI                                             *
* DMU-FOLU, 910205, Erik Runge                                           *
*                                                                        *
* Calculates the NOx emission on the main shiproutes through Danish      *
* waters.                                                                *
* The emission factors used are as read from Figure 7.5, in "EMEP        *
* workshop on emission from ships", Oslo, Norway 7-8 June 1990,;         *
* Bremnes, P.K. - "Calculations of exhaust gas emissions from sea        *
* transport. Methodology and results".                                   *
* The emission factors used are;                                         *
* Route T - from Skagen till southwest of Lolland (where the ships       *
* from the Kieler Canal joins with route T) -                            *
* 34 tonne NOx/year/nautic mile.                                         *
* Route T - from soute-west of Lolland into Østersøen -                  *
* 86 tonne NOx/year/nautic mile                                          *
* Route D - B - 21 tonne NOx/year/nautic mile                            *
*                                                                        *
* One nautic mile is equal to 1852 m.                                    *
*                                                                        *
* Call; NOX_EMI(DIST,NO,EMI)                                             *
* Input;  DIST  - The distance in km on which the emission occures       *
*         NO    - A number refering to the shiproute                     *
* Output; EMI   - The calculated emission on the distance DIST           *
**************************************************************************
c
      subroutine NOX_EMI(DIST,NO,EMI)
      integer NO
      real*8 DIST,EMI
c
      if (NO .eq. 1) EMI = DIST/1.852 * 34.
      if (NO .eq. 2) EMI = DIST/1.852 * 86.
      if (NO .eq. 3) EMI = DIST/1.852 * 21.
      if (NO .ne. 1 .and. NO .ne. 2 .and. NO .ne. 3)
     1 call FAIL(13,' Your have called NOX_EMI with a wrong
     1 value in NO',DIST,NO)
      return
      end
c
c
**************************************************************************
* Fortran subroutine GRID                                                *
* DMU-FOLU, 910207, Erik Runge                                           *
*                                                                        *
* Grids calculated emissions on a UTM grid.                              *
*                                                                        *
* Call: GRID(X_OLD,Y_OLD,X,Y,EMI)                                        *
* Input: X_OLD,Y_OLD - The UTM X- and Y- coordinate of the starting      *
```

```
*                        point in km.                                *
*        X,Y           - The UTM X- and Y- coordinate of the ending  *
*                        point in km.                                *
*        EMI           - The emission on the distance from X_OLD,Y_OLD *
*                        to X,Y                                      *
* Output: The gridded emissions are written in common array OUT      *
*                                                                    *
**********************************************************************
      subroutine GRID(X_OLD,Y_OLD,X,Y,EMI)
c
      integer MINX,MINY
      real*8  BEGX,ENDX,STEPX,X,X_OLD,
     1        BEGY,ENDY,STEPY,Y,Y_OLD,
     1        EMI,PEMI,OUT,T,J1
      common  OUT(570:770,6030:6420)
c
      BEGX = X_OLD
      ENDX = X
      BEGY = Y_OLD
      ENDY = Y
      J1   = 0.
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Determine the southwestern corner of the grid element, as a        *
* reference to that grid element.                                    *
* Note that MINX and MINY are integers.                              *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      MINX = BEGX
      MINY = BEGY
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the actual distance is not inside one gridelement, then step    *
* until the reference point is not longer the same.                  *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      if (int(ENDX) .ne. MINX .or. int(ENDY) .ne. MINY) then
      do 200 T = 1.,100000.,1.
         STEPX = BEGX + T/100000. * (ENDX - BEGX)
         STEPY = BEGY + T/100000. * (ENDY - BEGY)
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Calculate the part of the emission in this gridelement and write   *
* the value in array OUT.                                            *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         if (int(STEPX) .ne. MINX .or. int(STEPY) .ne. MINY) then
             PEMI = EMI * (T-J1)/100000.
             OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
             MINX   = STEPX
             MINY   = STEPY
             J1 = T
         endif
  200 continue
      endif
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the distance or the rest of the distance after stepping, is     *
* inside one grid element, then calculate the emission and write     *
* the value in array OUT.                                            *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      if (int(ENDX) .eq. MINX .and. int(ENDY) .eq. MINY) then
          PEMI = EMI * (1 - J1/100000.)
           OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
      endif
      return
      end
c
c
**********************************************************************
*                                                                    *
*   Fortran subroutine FAIL                                          *
*   DMU-FOLU, 910117, Erik Runge.                                    *
*                                                                    *
```

```
*   Writes a message to a file if an call of a subroutine        *
*   returnes with an unexpected answer                           *
*                                                                *
*   Call: FAIL(F_FILE,TEXT,A,B)                                  *
*                                                                *
*   Input:                                                       *
*     F_FILE - The file to which the message is written          *
*     TEXT   - The text written to file F_FILE                   *
*     P1,P2  - Paramters,                                        *
*              used to indicate where the error occured          *
*                                                                *
******************************************************************
c
      subroutine FAIL(F_FILE,TEXT,P1,P2)
c
      integer F_FILE
      real P1,P2
      character*(*) TEXT
c
         write(F_FILE,130)TEXT,P1,P2
 130     format(a,2f15.4)
c
       return
      end
```

```
************************************************************************
* Fortran program FERRY                                                *
* 910205, DMU-FOLU, Erik Runge                                         *
*                                                                      *
* This program calculates the NOx emission, in tonne NOx/year, on the  *
* main ferryroutes through Danish waters. The routes are;              *
* 1 - Rødby       - Puttgarden                                         *
* 2 - Kalundborg - Samsø                                               *
* 3 - Bøjden      - Fynshavn                                           *
* 4 - Gedser      - Warnemunde                                         *
* 5 - Korsør      - Nyborg                                             *
* 6 - Halskov     - Knudshoved                                         *
* 7 - Kalundborg - Århus                                               *
* 8 - København  - Helsingborg                                         *
* 9 - Helsingør  - Helsingborg                                         *
*                                                                      *
* The calculated emissions are distributed on a 1x1 km2 UTM Zone 32    *
* grid, for the use in dispersion models.                              *
*                                                                      *
* The variables;                                                       *
*    DIST  - The distance in meters                                    *
*    EMI   - The calculated emission on actual route or part of route. *
*    NEW   - Values used in Uniras plotting program, 2 ~ 'hold         *
*            pen down' (one continous line), 3 ~ 'lift pen up and      *
*            put it down'.                                             *
*    NO    - A number of the route, used to determine which emis-      *
*            sion factor is to be used.                                *
*    X_OLD,Y_OLD,X,Y - The starting and ending points of part of the   *
*            shiproute.                                                *
* Arrays;                                                              *
*    DIST  - Holds for each of the nine routes the distance in km.     *
*    OUT   - Holds the calculated NOx emission each UTM Zone 32        *
*            grid elements.                                            *
* Files;                                                               *
*    File 10 - Holds the coordinates, in UTM Zone 32, to the ship      *
*              routes.                                                 *
*    File 13 - The 'error' file to which errors - if any - is          *
*              written during program execution.                       *
*    File 20 - Is the 'output' file, to which the calculated NOx       *
*              emissions are written.                                  *
*                                                                      *
* Subroutines called;                                                  *
*    Subroutine NOX_EMI                                                *
*    Subroutine GRID                                                   *
*    Subroutine FAIL                                                   *
*                                                                      *
************************************************************************
c
      program FERRY
c
      integer NEW,NO
      real*8 DIST,P_DIST,EMI,OUT,X,X_OLD,Y,Y_OLD
      common OUT(563:732,6008:6224),DIST(1:9)
c
      open(10,file='msluft$met:[havdata.kortplot]MAIN_FERRIES.DAT',
     1        readonly,status='OLD')
c
      open(13,file='msluft$met:[havdata.fortran]FERRY.ERR',
     1        status='NEW',carriagecontrol='LIST')
c
      open(20,file='msluft$met:[havdata.data]FERRY_NOX_EMI.DAT',
     1        status='NEW',carriagecontrol='LIST')
c
      do  60 I = 563,732,1
         do 50 J = 6008,6224,1
            OUT(I,J) = 0.
 50      continue
 60   continue
c
      read(10,110,end=998)X,Y,NEW,NO
400   continue
```

```
       X_OLD = X
       Y_OLD = Y
       read(10,110,end=998)X,Y,NEW,NO
       if (NEW .eq. 2) then
          DIST(NO) = DIST(NO) + sqrt((X-X_OLD)**2 + (Y-Y_OLD)**2)
          goto 400
       else
          goto 400
       endif
c
  998 continue
      rewind 10
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Read a record, if it is part of the same route (NEW = 2) then      *
* calculate and grid the emission. If the record read is start of    *
* a new route then save the X,Y coordinates and read the next record. *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  500  continue
       X_OLD = X
       Y_OLD = Y
       read(10,110,end=999)X,Y,NEW,NO
       if (NEW .eq. 2) then
          P_DIST = sqrt((X-X_OLD)**2 + (Y-Y_OLD)**2)
          call NOX_EMI(P_DIST,NO,EMI)
          if (EMI .le. 0.) call FAIL(13,'EMI is .le. 0 after NOX_EMI',
     1    P_DIST,EMI)
          call GRID(X_OLD,Y_OLD,X,Y,EMI)
          goto 500
       else
          goto 500
       endif

  110  format(1x,2f10.5,i3,26x,i2)
  999  continue
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Write the gridded calculated emissions to external file            *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       do  30 I = 563,732,1
          do 40 J = 6008,6224,1
             if (OUT(I,J) .ne. 0.) then
c             sum = sum + OUT(I,J)
             write(20,220) I,J,OUT(I,J)
  220        format(i4,i5,f15.4)
            endif
   40     continue
   30  continue

c       write(6,*) sum
       end
c
c
**********************************************************************
* Fortran subroutine NOX_EMI                                         *
* DMU-FOLU, 910205, Erik Runge                                       *
*                                                                    *
* Calculates the NOx emission on the main ferryroutes through Danish *
* waters.                                                            *
*                                                                    *
* Call; NOX_EMI(DIST,NO,EMI)                                         *
* Input;  DIST  - The distance in km on which the emission occures   *
*         NO    - A number refering to the shiproute                 *
* Output; EMI   - The calculated emission on the distance DIST       *
*                 In tonne NOx/year                                  *
* Variables; DENSI - The density of the fuel used, here has not been *
*            taken into account the different densities of different *
*            fuels. Value in ton/m3                                  *
*            EMIFAC - The emission factor in kg NOx/ton. Read from;  *
*            'EMEP workshop on emissions from ships', Oslo, 1990     *
**********************************************************************
```

```
c
      subroutine NOX_EMI(P_DIST,NO,EMI)
      integer NO,OIL
      real*8 DENSI,DIST,P_DIST,EMI,EMIFAC,OUT
      common OUT(563:732,6008:6224),DIST(1:9)
      dimension OIL(1:9)
c
      data OIL/24384,1519,757,298,27619,33292,23035,7600,10100/
      DENSI  =  0.85
      EMIFAC = 70.
c
      EMI = OIL(NO)/DIST(NO) * P_DIST * DENSI * EMIFAC/1000.
c
      return
      end
c
c
***********************************************************************
* Fortran subroutine GRID                                             *
* DMU-FOLU, 910207, Erik Runge                                        *
*                                                                     *
* Grids calculated emissions on a UTM grid.                           *
*                                                                     *
* Call: GRID(X_OLD,Y_OLD,X,Y,EMI)                                     *
* Input: X_OLD,Y_OLD - The UTM X- and Y- coordinate of the starting   *
*                      point in km.                                   *
*        X,Y         - The UTM X- and Y- coordinate of the ending     *
*                      point in km.                                   *
*        EMI         - The emission on the distance from X_OLD,Y_OLD  *
*                      to X,Y                                         *
* Output: The gridded emissions are written in common array OUT       *
*                                                                     *
***********************************************************************
      subroutine GRID(X_OLD,Y_OLD,X,Y,EMI)
c
      integer MINX,MINY
      real*8  BEGX,ENDX,STEPX,X,X_OLD,
     1        BEGY,ENDY,STEPY,Y,Y_OLD,
     1        EMI,PEMI,OUT,T,J1,DIST
      common OUT(563:732,6008:6224),DIST(1:9)
c
      BEGX = X_OLD
      ENDX = X
      BEGY = Y_OLD
      ENDY = Y
      J1   = 0.
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Determine the southwestern corner of the grid element, as a       *
* reference to that grid element.                                    *
* Note that MINX and MINY are integers.                              *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      MINX = BEGX
      MINY = BEGY
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the actual distance is not inside one gridelement, then step   *
* until the reference point is not longer the same.                 *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      if (int(ENDX) .ne. MINX .or. int(ENDY) .ne. MINY) then
      do 200 T = 1.,100000.,1.
         STEPX = BEGX + T/100000. * (ENDX - BEGX)
         STEPY = BEGY + T/100000. * (ENDY - BEGY)
c
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Calculate the part of the emission in this gridelement and write  *
* the value in array OUT.                                            *
*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         if (int(STEPX) .ne. MINX .or. int(STEPY) .ne. MINY) then
             PEMI = EMI * (T-J1)/100000.
             OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
```

```
              MINX    = STEPX
              MINY    = STEPY
              J1 = T
           endif
  200  continue
       endif
c
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* If the distance or the rest of the distance after stepping, is     *
* inside one grid element, then calculate the emission and write     *
* the value in array OUT.                                            *
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       if (int(ENDX) .eq. MINX .and. int(ENDY) .eq. MINY) then
           PEMI = EMI * (1 - J1/100000.)
            OUT(MINX,MINY) = OUT(MINX,MINY) + PEMI
       endif
       return
       end
c
c
**********************************************************************
*                                                                    *
*   Fortran subroutine FAIL                                          *
*   DMU-FOLU, 910117, Erik Runge.                                    *
*                                                                    *
*   Writes a message to a file if an call of a subroutine            *
*   returnes with an unexpected answer                               *
*                                                                    *
*   Call: FAIL(F_FILE,TEXT,A,B)                                      *
*                                                                    *
*   Input:                                                           *
*     F_FILE - The file to which the message is written              *
*     TEXT   - The text written to file F_FILE                       *
*     P1,P2  - Paramters,                                            *
*              used to indicate where the error occured              *
*                                                                    *
**********************************************************************
c
       subroutine FAIL(F_FILE,TEXT,P1,P2)
c
       integer F_FILE
       real P1,P2
       character*(*) TEXT
c
          write(F_FILE,130)TEXT,P1,P2
  130     format(a,2f15.4)
c
        return
       end
```

Appendix B

Format of output files

In the following the format of the output files are given as read and written by fortran.

The file, NOX_AREA, with emissions from area sources has the format:

```
i3,i5,6(1x,f10.5)
```

with the collumns:

X and Y coordinate (UTM zone 32)
Emission (tonne NO_2 a^{-1}) from:
domestic heating,
energy generation,
combustion in industry,
industrial production,
road traffic and
total emission

The file, NOX_POINT, with emissions from point sources has the format:

```
a20,1x,i2,i7,i8,2i4,i8,i6
```

with the collumns:

Name of point source
X and Y coordinate (global longitude and latitude)
Height (m) of chimney
Temperature (°C) of flue gas
Volume (1000 m^3 a^{-1})
Emission (Tonne NO_2 a^{-1})

The files, NOX_FREIGTHER and NOX_FERRY, with emissions from freigther and ferry traffic have the format:

```
i4,i5,f15.4
```

with the collumns:

X and Y coordinates (UTM zone 32)
Emission (tonne NO_2 a^{-1})

Bibliographic Data Sheet **Risø-M-2929**

Title and author(s)

A detailed Emission Inventory of Nitrogen Oxides for Denmark

Erik H. Runge, Willem A.H. Asman and Niels A. Kilde

ISBN	ISSN
87-550-1730-4	0418-6435

Dept. or group	Date
DMU and System Analysis Department	August 1991

Groups own reg. number(s)	Project/contract no.

Pages	Tables	Illustrations	References
68	5	13	9

Abstract (Max. 2000 characters)

The Danish NO_x emissions from domestic heating, energy generation, industry, road traffic, point sources and maritime vessels have been distributed on a 1x1 km^2 grid and on municipalities.

The total NO_x emission is calculated to 294 ktonne NO_2 a^{-1}.

This report describes the distribution, lists the computer programs used and gives the format of the final storage file.

Descriptors INIS/EDB

COMPILED DATA; DENMARK; EMISSION; FOSSIL-FUEL POWER PLANTS; HEATING; INDUSTRY; INVENTORIES; NITROGEN OXIDES; POINT POLLUTANT SOURCES; POLLUTION SOURCES; POWER GENERATION; REGIONAL ANALYSIS; RESIDENTIAL BUILDINGS; ROAD TRANSPORT; SHIPS; SPATIAL DISTRIBUTION; STATISTICAL DATA; VEHICLES

Available on request from Risø Library, Risø National Laboratory, (Risø Bibliotek, Forskningscenter Risø), P.O.Box 49, DK-4000 Roskilde, Denmark. Telephone 42 37 12 12, ext. 2268/2269 Telex 43 116. Telefax 46 75 56 27.